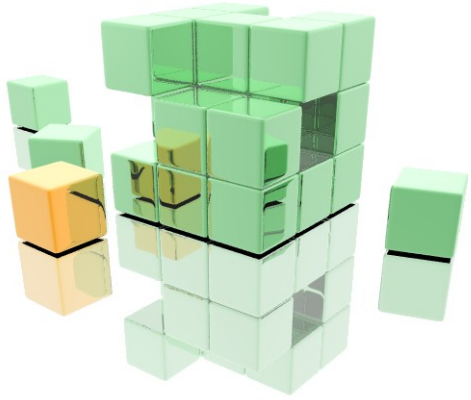


tcACCESS Version 8
The Windows Client

Last review: 2010-04-28

TABLE OF CONTENTS

Table of figures

1. Installing tcACCESS

1.1 General

To install tcACCESS, run the SETUP.EXE program from the tcACCESS CD.

You may use UNC names without any restrictions when you install the program. If you are installing the program on a server and the users are connected to different disk drives, we recommend that you use UNC names, e.g., use \\SERVER\COMMON\tcACCESS instead of X:\tcACCESS.

Note: UNC is short for Universal Naming Convention, a PC format for specifying the location of resources on a local-area network (LAN).

1.2 Installation Types

The following table describes the tcACCESS installation options :

INSTALLATION TYPE	DESCRIPTION
Full	All PC components, including files and settings, are installed locally. You may elect to install the libraries for the host components locally as well. You should use a full installation when an MS Windows network is not permanently available for the PCs to avoid additional network load.
Server preparation	The server preparation creates a directory structure identical to a local installation on the server. All language files are copied during a server preparation. **Note:** The server preparation is the first step to a network installation. When the server preparation is complete, each client PC must be connected to this server preparation.
Link to a server installation	When the server preparation is complete, you can create a connection to the server for the clients. To create the client connection, run the SETUP.E program again. When you are creating the connection to the server, no files are stored locally. The programs and setting files are accessed via the server. The advantage of this type of installation is that all settings are made once on the server, with all client PCs having the same available settings. Also, you can install product updates on the server from an optional client.
Upgrade	If tcACCESS version 3.0 is already installed it can be upgraded by installing the 4.0 version. The programs are updated while the settings are preserved.

INSTALLATION TYPE	DESCRIPTION
	This switch will not being used when upgrading from a Version 5.0.

If the installation media contains different license types (i.e. Server-, Administrator- or special Client-licenses), a dialog will be displayed to select the desired license type. It is possible to perform an installation based upon the license-type.

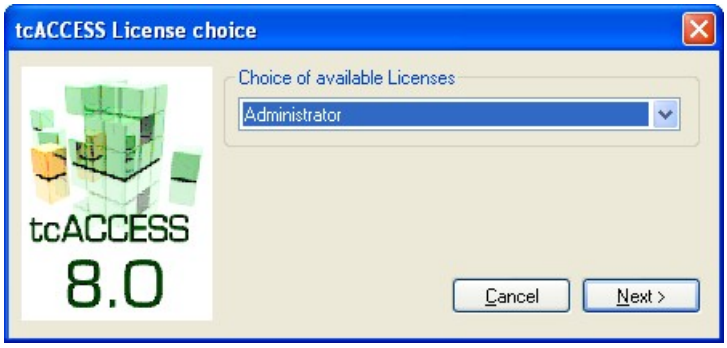


Fig. 1:Selection of installation licenses

This dialog will not be displayed if key "LicNr" has been set in file "SETUP.INI".

1.3 Full Installation

To perform a full installation:

1. Run the SETUP.EXE program on the tcACCESS installation CD.

   Note: When the MainInstall key is not set in the SETUP.INI file, the installation displays the Type of installation dialog window, shown in Fig. 1.

2. Select FULL INSTALLATION to perform a local installation.

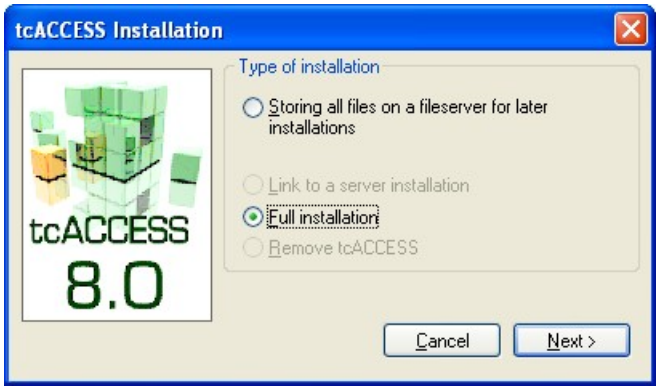


Fig. 2: Full Installation - Start dialog

3. Click the **Next** button.

⇒ The tcACCESS directory window displays.

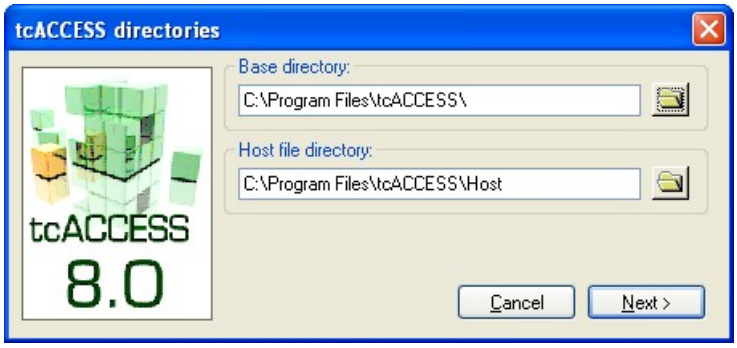


Fig. 3: Full Installation - Start dialog

Note:

- You can use the **BaseDir** key in the SETUP.INI file to define the base directory for the installation. If no setting is made, the base directory is defined automatically by the default value "C:\Program Files\tcACCESS".

- The "C:\Program Files" directory represents the program directory defined under MS Windows. Use the base directory dialog window to select the directory. This dialog window displays only when the **NoDirDlg** key is not set to **Y** in the SETUP.INI file.
- If the base directory of the installation is changed, the system modifies all other directory settings accordingly. To change a directory, click the browse button next to the directory type that you want to change.

4. Click the **Next** button.

⇒ The tcACCESS Options window displays.

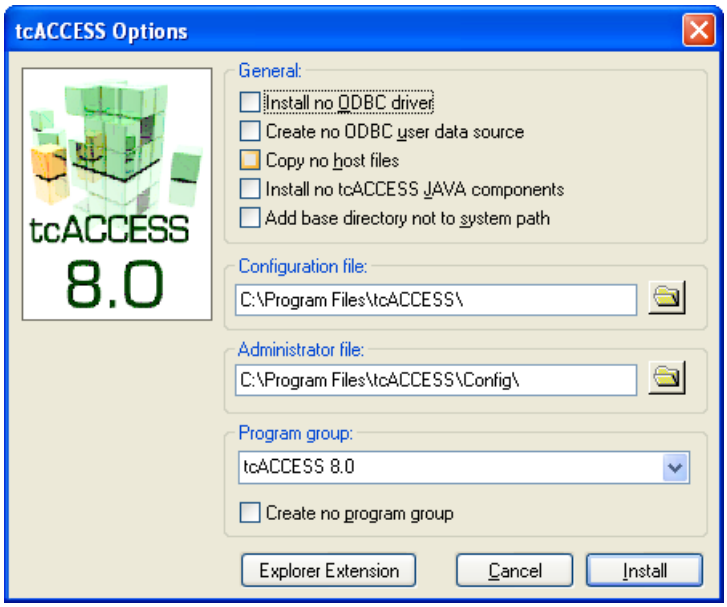


Fig. 4: Full Installation – Options

5. Update each option as required. Refer to a description of each option below.

Note: This window does not display when the NoOptionsDlg key is set to Y in the SETUP.INI file. In this case, the system accepts all settings based on the SETUP.INI default values.

OPTION	DESCRIPTION
Install no ODBC driver	If you select this option, the system does not install the ODBC driver. This option corresponds to the **NoODBC** key in the SETUP.INI file.
Create no ODBC user data source	After you install the ODBC driver, the installation program asks for the user data source definition. You can suppress this request by activating the **Do not define user data resource** key. This option corresponds to the **NoUserDSN** key in the SETUP.INI file. If this option has been used, no DSN entries will be created. However, both drivers (ANSI and UNICODE) will be installed.
Copy no host files	If you select this option, the system does not copy the host components. The option corresponds with the **NoHost** key in the SETUP.INI file.
Install no tcACCESS JAVA components	If you select this option, the system will not install the JDBC driver and tcACCESS JListener. Both are necessary to access databases from a JAVA program. This option corresponds to the **NoJAVA** key in the SETUP.INI file.
Add base directory not to system path	If you select this option, the system does not add the tcACCESS base directory to the system path. This option corresponds to the **NoSysPath** key in the SETUP.INI file. A system path entry is not necessary for a normal program ecution. Request this entry when you develop in-house programs that use the API to load the TCA.DLL dynamically.
Configuration file	This directory determines the location for the local tcACCESS.INI configuration file. The MS Windows directory is the default directory.
Administrator file	This directory determines the location of the settings files for the administrator. This applies to the main configuration file "tcACCESS.cfg" and the configuration files for the connection slots (extension *.slc). The "CONFIG" directory is the default setting.
Program group	You can enter a program group name or select an existing program group. The default value corresponds to the GroupName setting in the SETUP.INI file.
Create no program group	This option suppresses the creation of a program group. The setting corresponds with the **NoApplGroup** key in the SETUP.INI file.
Explorer Extension	When you click this button, a dialog box displays

OPTION	DESCRIPTION
	and lets you define the Explorer Extension installation. You can change the location of the extension between Desktop and My Computer.

6. Click the **Next** button.

   ⇒ A status window displays the progress of the installation.

Note: The Cancel button does not display when the NoCancel key in the SETUP.INI file is set to Y.

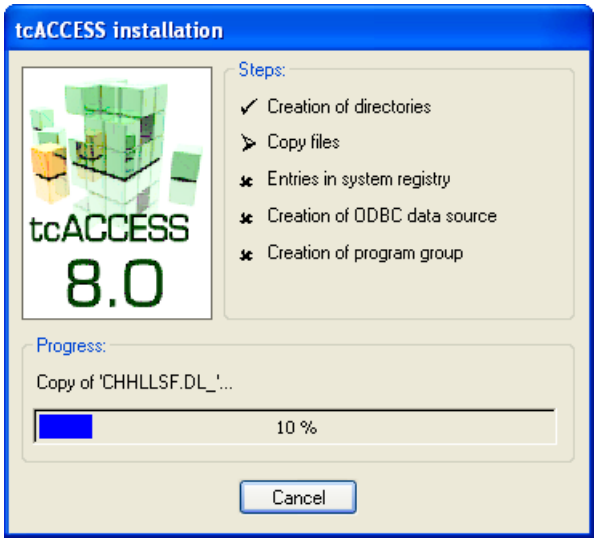


Fig. 5: Full Installation - Status window

The following table describes the progress symbols:

SYMBOL	DESCRIPTION
✓	The installation step has completed successfully.
✖	The installation step has not completed successfully or has been skipped due to the selected options.
➢	The installation step is currently in progress.

Notes:

- If the system does not detect the basic ODBC installation on a PC, the system does not install the ODBC driver. In this case, a message displays.
- If the existing basic ODBC installation is not up-to-date, the system updates it automatically.

7. Create a user data source.

   ⇒ The system displays this request, provided this step is not suppressed. A detailed description for the configuration of a user data resource can be found in section The tcACCESS ODBC driver, page. *369*

When the installation is complete, the system defines the program group. The program group allows the user to start individual programs from the **Start** menu.

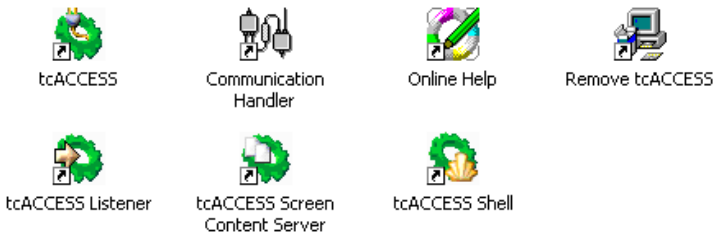


Fig. 6: Full Installation - tcACCESS Program Group

1.4 Server Installation

You must perform a server preparation before a network installation. To perform a server preparation:

1. Run the SETUP.EXE program on the tcACCESS installation CD.

Note: When the MainInstall key is not set in the SETUP.IN file, the installation displays the Type of installation dialog window, shown in the figure below:

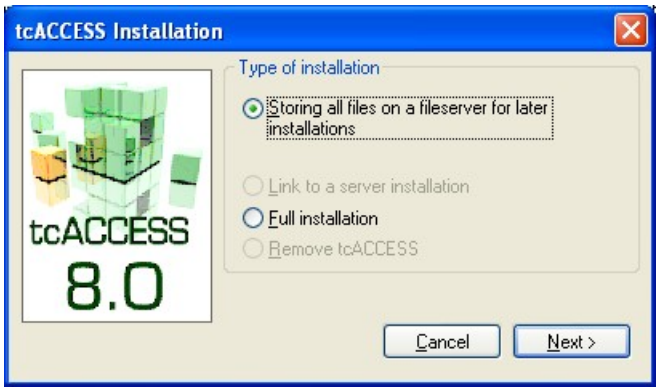


Fig. 7: Server Preparation - Start dialog

2. Select PREPARATION OF A SERVER option to perform a network installation.

3. Click the **Next** button.

⇒ The tcACCESS directory window displays.

Note:

- You can use the **BaseDir** key in the SETUP.INI file to define the base directory for the installation. If no setting is made, the base directory is defined automatically by the following default value:

  \\???\???\tcACCESS

- This dialog window displays only when the **NoDirDlg** key is not set to **Y** SETUP.INI file. During server preparation, you must provide a base directory using the **BaseDir** key.

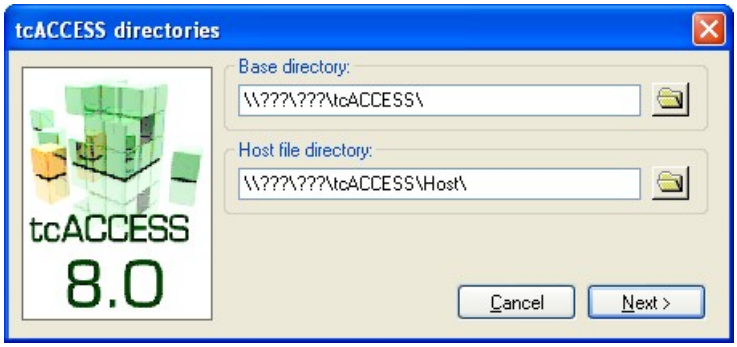


Fig. 8: Server Preparation - Directory selection

4. Click the **Next** button.

   ⇒ A status window displays the progress of the installation.

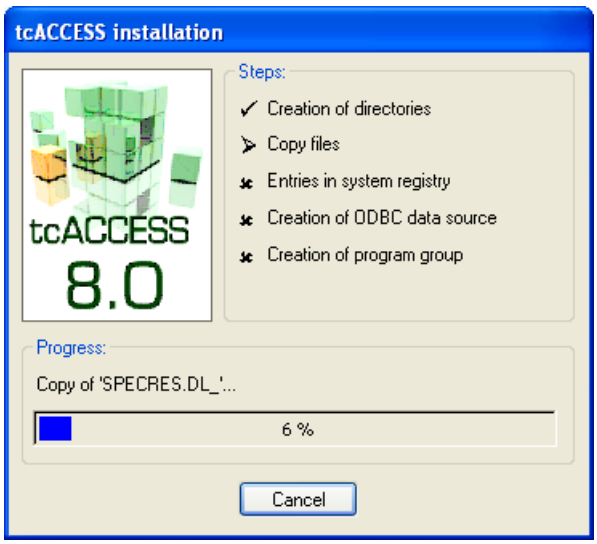


Fig. 9: Server Preparation - Status window

The following table describes the progress symbols:

SYMBOL	DESCRIPTION
✓	The installation step has completed successfully.
✖	The installation step has not completed successfully or has been skipped due to the selected options.
➢	The installation step is currently in progress.

Note:

- To install a client from a prepared server, you must start the installation program from the server.
- You may use UNC names when you install the program without any restrictions. If you are installing the program on a server and the users

are connected to different disk drives, we recommend that you use UNC names, e.g., use \\SERVER\COMMON\tcACCESS instead of X:\tcACCESS.

1.5 Installing a Client

When you install a client (link to a server preparation), the system does not install the PC components locally.

Note: You must perform a server preparation before a network installation. To perform a server preparation:

1. Run the SETUP.EXE program on the tcACCESS installation CD.

   Note: When the MainInstall key is not set in the SETUP.INI file, the installation displays the Type of installation dialog window, shown in Fig. 9.

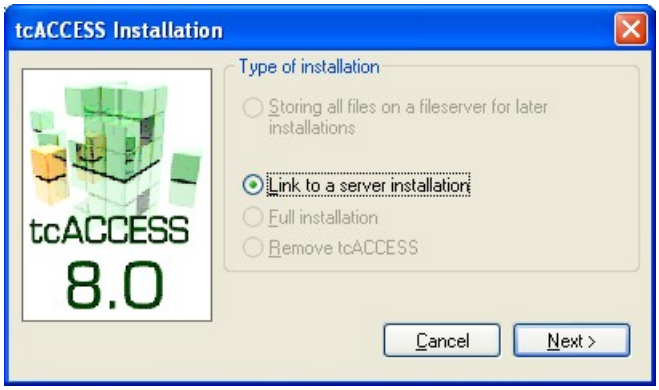


Fig. 10*: Client Installation – Start dialog*

2. Click the **Next** button.

   ⇒ The tcACCESS Options window displays.

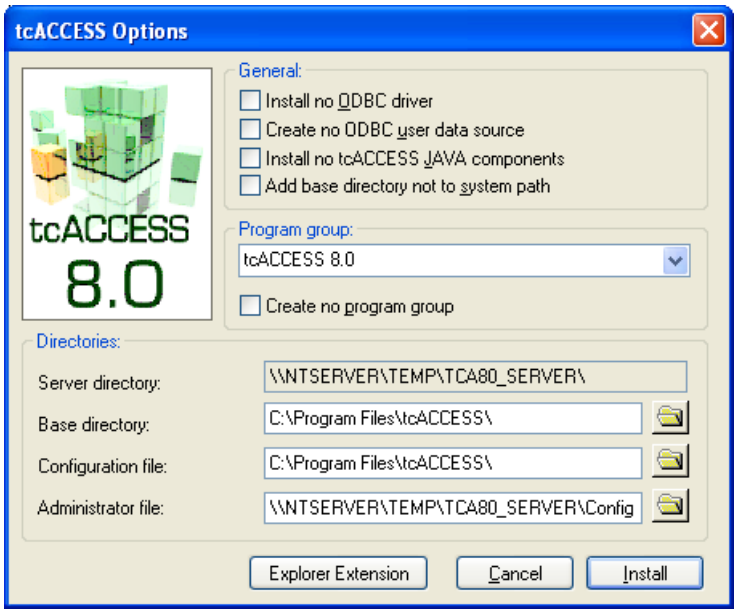


Fig. 11: Client Installation – Options

3. Update each option as required. Refer to a description of each option below.

Note: This window does not display when the NoOptionsDlg key is set to Y the SETUP.INI file. In this case, the system accepts all settings based on the SETUP.INI default values.

OPTION	DESCRIPTION
Install no ODBC driver	If you select this option, the system does not install the ODBC driver. This option corresponds to the **NoODBC** key in the SETUP.INI file.
Create no ODBC user data source	After you install the ODBC driver, the installation program asks for the user data source definition. You can suppress this request by activating the **Do not define user data resource** key. This option corresponds to the **NoUserDSN** key in the SETUP.INI file.
Install no tcACCESS JAVA components	If you select this option, the system does not install the JDBC driver and tcACCESS Jlistener. Both are necessary to access databases from a JAVA program. This option corresponds to the **NoJAVA** key in the SETUP.INI file.
Add base directory not to system path	If you select this option, the system does not add the tcACCESS base directory to the system path. The option corresponds to applying the **NoSysPath** key in the SETUP.INI file. A system path entry is not necessary for a normal program ecution. Request this entry when you develop in-house programs that use the API to

OPTION	DESCRIPTION
	load the TCA.DLL dynamically.
Program group	You can enter a program group name or select an existing program group. The default value corresponds to the GroupName setting in the SETUP.INI file.
Create no program group	This option suppresses the creation of a program group. The setting corresponds with the **NoApplGroup** key in the SETUP.INI file.
Server directory	This field shows the server directory that the client is connected to after the installation. All programs, modules, and language files are located on this directory. A tcACCESS user does not require write authorization for this directory.
Base directory	This field shows the user specific directories and files on the local PC. A tcACCESS user must have write authorization for these local directories.
Configuration file	This directory determines the location for the local tcACCESS.INI configuration file. The MS Windows directory is the default directory.
Administrator file	This directory determines the location of the settings files for the administrator. This applies to the main configuration file "tcACCESS.cfg" and the configuration files for the connection slots (extension *.slc). The "CONFIG" directory is the default setting.
Explorer Extension	When you click this button, a dialog box displays and let's you define the Explorer Extension installation. You can change the location of the extension between Desktop and My Computer.

4. Click the **Next** button.

   ⇒ A status window displays the progress of the installation.

   Note: The Cancel button does not display when the NoCancel key in the SETUP.INI file is set to Y.

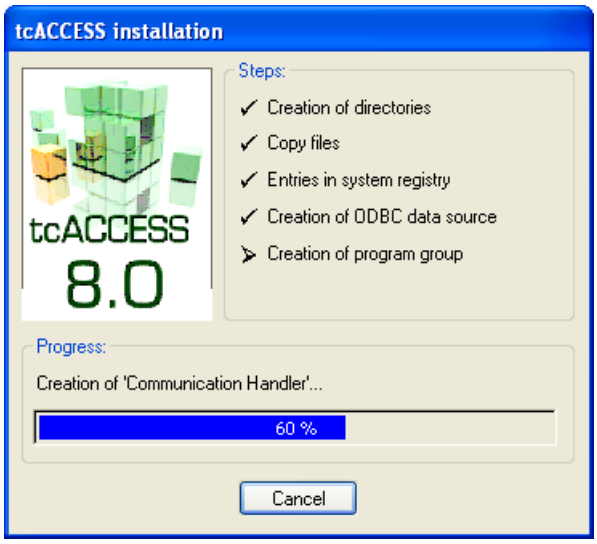


Fig. 12: Client Installation - Status Window

The following table describes the progress symbols:

SYMBOL	DESCRIPTION
✓	The installation step has been completed successfully.
✖	The installation step has not been completed successfully or has been skipped due to the selected options.
➢	The installation step is currently in progress.

Note: If the system does not detect the basic ODBC installation on a PC, the system does not install the ODBC driver. In this case, a message displays.

5. Create a user data source.

   ⇒ The system displays this request, provided this step is not suppressed. A detailed description for the configuration of a user data resource can be found in section *The tcACCESS ODBC driver*, page *369*.

When the installation is complete, the system defines the program group. The program group allows the user to start individual programs from the Start menu.

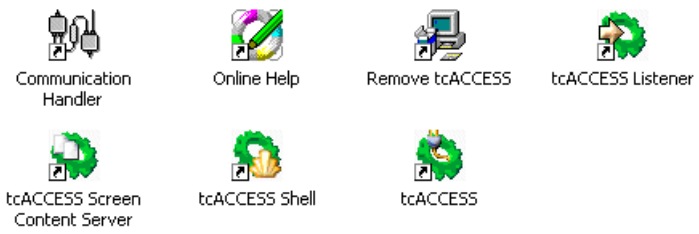


Fig. 13: Client Installation - tcACCESS Program Group

1.6 Predefined Installation

Run the SETUP.E program on the tcACCESS installation CD. **Note**: When you specify parameter entries for the SETUP.INI file, the system processes the installation without any user intervention.

You can use the following entry as command line parameters.

/uninstall tcACCESS is uninstalled.

You can make additional specifications in the SETUP.INI file. The file format is equivalent to an MS Windows configuration file (*.INI). You can customize the file based on individual requirements by using any editor. You enter all settings in the [Main] section.

KEY	DEFAULT VALUE	DESCRIPTION AND POSSIBLE SETTINGS
MainInstall	DIALOG	• DIALOG User-controlled installation by dialogs • FULL Local full installation • SERVER Preparation of a server • CLIENT Connection to a server installation • DEINSTALL tcACCESS is uninstalled
BaseDir	C:\ProgramFiles\tcACCESS	Base directory for a local installation, which includes the following sub-directories: • WORK Work directory • PARAM Parameter directory • CONFIG Configuration files • SAMPLES Sample directory • EXPWxx Online help binaries (only for MainInstall=FULL or MainInstall=SERVER) • LNG Directory with language dependent files
Language	D	Language identifier for the product. The following languages are currently available: • D German • E English (UK)

KEY	DEFAULT VALUE	DESCRIPTION AND POSSIBLE SETTINGS
		• A English (USA) • I Italian • F French
GroupName	tcACCESS 6.0	Name of the program group that is created by the installation program (only when MainInstall=FULL or MainInstall=CLIENT)
IniPath	tcACCESS Install. path	If this parameter is defined, all configuration settings are stored at the specified location.
CfgPath	Base installation directory + "\CONFIG"	If this parameter is defined, the system stores the administrator settings and the connection settings at the specified location.
NoApplGroup	N	If this parameter is set to **Y**, the system does not create a group entry on the Start menu.
DesktopIcon	N	If this parameter is set to **Y**, the system creates a link on the desktop (Explorer Shell) for the program group.
NoODBC	N	If this parameter is set to **Y**, the system does not install the ODBC driver.
NoJAVA	N	If this parameter is set to **Y**, the system does not install tcACCESS JAVA components.
NoCancel	N	If this parameter is set to **Y**, the **Cancel** button on the status display is inactive.
NoHostFiles	N	If this parameter is set to **Y**, the system does not copy the files with the host components.
NoSysPath	Y	If this parameter is set to **Y**, the installation directory extends the system path. WIN 95/98 and 3.11 require a system reboot.
NoUserDSN	N	If this parameter is set to **Y**, the system does not request a user data source.
NoOptionsDlg	N	If this parameter is set to **Y**, the tcACCESS Options window does not display. All values deviating from the default values must be specified in the SETUP.INI.
NoDirDlg	N	If this parameter is set to **Y**, the tcACCESS Directory window does not display. The base installation directory is defined with the **BaseDir** key; all other directories listed above are equivalent to the default values.
NoShellExt	Y	If this parameter is set to **Y**, the system does not install the Explorer Extension.
ExtOnMy Computer	N	If this parameter is set to **Y**, the system installs Explorer Extension under My

KEY	DEFAULT VALUE	DESCRIPTION AND POSSIBLE SETTINGS
		Computer and not Desktop.
NoDelete	N	If this parameter is set to **Y,** the directories which have been created during a client installation, will not be deleted.
Upgrade40	N	If this parameter is set to **Y**, the system attempts to upgrade from version 4.0. This only applies for a full installation.
LicNr	0	This key is only valid, if the installation media contains multiple license types (i.e. Server and Administrator). This key defines that a certain PC-license should be transferred from file SETUP.INI. If the key is not defined a dialog box will be displayed to select the appropriate license.
ODBC20	N	This parameter specifies whether the ODBC data-source should return "2.0" as version information.
NoStpCompat	N	A specification of "Y" ensures that the tcACCESS SQL Stored Procedure calls received from the tcACCESS ODBC data source accept the parameter syntax from earlier tcACCESS versions. In this case only the input parameter must be supplied when calling a SQL-Engine Stored procedure.

1.7 ODBC Data Source Installation

This version of the installation program allows you to install ODBC data sources without user intervention. Define each data source to be installed in the section titled ODBCDSNx located in the SETUP.INI file. The x stands for a number, which differentiates the individual sections. If you install more than one data source, the sections must be numbered consecutively, e.g., ODBCDSN1, ODBCDSN2, ODBCDSN3, etc.

Within the individual sections that you create, use the following key words: The parameters described below are explained in detail in chapter The tcACCESS ODBC driver, page *369*.

KEY	DEFAULT VALUE	DESCRIPTION AND POSSIBLE SETTINGS
System	No	If this parameter is set to **Yes**, the data source is installed as a system data source and is available for each user on the workstation.
Name		The name of the data source must be specified. A data source without a name terminates the installation of DSNs.
Description		Enter a description in this field for the data source.
UID		A default signon name can be defined to establish the connection.
PWD		A default password can be assigned to establish the connection.
Parameter		The data source parameters are described in chapter The tcACCESS ODBC driver, page 369.
Trace	No	If this parameter is set to Yes, the ODBC accesses are logged.
ShowStat	No	A status window showing access information displays if this option is set to Yes.
Type	TCA/SQLX	This parameter identifies the host database management system to be used. The following options are available: • TCA/SQLX tcACCESS SQL-Engine • SQL/DS SQL/DS • DB2 DB/2
Index	No	This option can only be selected for the host database types DB2 and SQL/DS (Yes). If it is set, tcACCESS delivers index information to the ODBC client using the batch files ODBCSCO.BAT (Special Columns) and

KEY	DEFAULT VALUE	DESCRIPTION AND POSSIBLE SETTINGS
		ODBCSTA.BAT (Statistic). By applying this information the ODBC client application can process INSERT, UPDATE and DELETE operations. However, the current user must have the sufficient mainframe authorizations. When host data and an ODBC client are incompatible, problems may occur. For example, the TIMESTAMP fields in DB2 contain six decimal positions for a fraction of split second. This is supported by Microsoft Query, but not by Microsoft Access. If a TIMESTAMP is used as primary key, a problem occurs with Microsoft Access when the fraction of a second value is unequal zero.

Transactions	No	This option is available for the host database types DB2, SQL/DS and tcACCESS SQL-Engine. For tcACCESS SQL-Engine databases the option is available only if logical transactions are supported by the database itself. For example: DL/I databases support logical transations. The same applies to VSAM files with CICS Journalling/Recovery. On the contrary, PS/PDS, SAM files or VSAM Batch files do not support logical transactions. If it is selected, the ODBC support is activated by logic transactions. It is the responsibility of the calling Windows application to use this support appropriately. If the support option is inactive, there is no ODBC support available for logic transactions. However, you can still submit the following standard ODBC SQL commands, for example, by using PASS THROUGH QUERIES in Microsoft Access: NOCOMMIT, COMMIT, ROLLBACK.
Systab	No	If this parameter is set to **Yes**, access to system data objects like *SQLTables* are handled with internal ODBC commands instead of calling external batch files, e.g., *ODBCTAB.BAT*. If special processing is necessary to access system objects, you should activate this option for performance reasons.

Creator	No	If this parameter is set to **Yes**, the name of the table creators are assigned as a prefix to each table name, which is returned to the calling program. It is separated by a dot (e.g., CREATOR.TABLE). In this case the creator information does not pass to the program (table without creator). This function is needed if the applied PC application does not use the name of the creator and therefore is not able to clearly address the tables residing on the host (e.g., for Microfocus NetExpress).
SendStatistics	No	If this parameter is set to **Yes**, after a data transfer the statistic is sent to the mainframe.
AllowDealloc-Session	No	If this option is set, a tcACCESS host session established by an ODBC driver is automatically terminated during a "SQLFreeConnect" (e.g. at termination of MS ACCESS). Sessions already established are not concerned, even if this option is set.
FixedFieldSize	No	If this option is set, trailing blanks of a field with type "string" are not removed.
UsePrepared-Commands	No	If this option is set to **Yes**, statements are prepared on the host and will be executed as "Prepared Commands".
HostConnection	Default	Specification of the ODBC DSN to be used for this connection. The following parameter can be specified: **"Default"** The defined slot or configuration will be used (Defined in the PC front end program under administrator settings). **"Slot"** The defined connection slot should be used. **"Config"** For the connection setup a slot will be used, which is specified using the CONFIG keyword.
Config		Configuration name, if HostConnection=Config
Slot	0	Connection Slot if "HostConnection=Slot"
OptimisticReadAhead	No	This key defines whether the new ODBC data-source should support Optimistic Read Ahead
LogonAsSqlId	No	This key specifies whether the new ODBC data-source should use the current username of the session sign-on as

		default SQLID.

1.8 Uninstall

You can uninstall tcACCESS by selecting the **Remove tcACCESS** symbol within the program group. After confirmation, the product is completely removed from the system. Depending on the type of installation, the following options are available:

(a) Full installation, where the directories and settings are removed.
(b) Client installation, where the local files and settings are removed.

Notes: Uninstalling a client installation is only effective in deleting local product files. The server preparation is not affected. If user data sources have been created based on the tcACCESS ODBC driver, they are removed also.

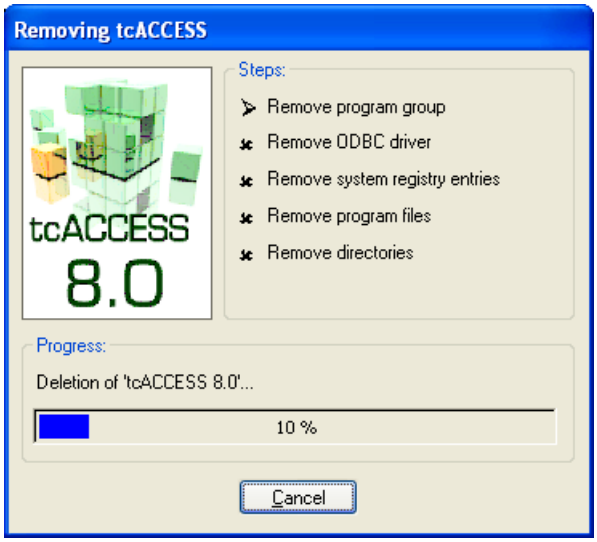


Fig. 14: Client Installation - Status Window

2. tcACCESS Windows Applications

The tcACCESS program provides interactive controls and configuration for your PC. tcACCESS does not configure the ODBC driver. The system uses Windows-standard dialog windows.

Notes for users of Microsoft Windows Vista:
Some functions of tcACCESS require administrator rights. As an example this includes the activation and deactivation of the tcACCESS Explorer Extension or the registration of the tcACCESS Listener as Service. In order to execute those functions, the corresponding program must once be executed with administrator permissions.
The tcACCESS files (configurationen, traces, parameter) can be found in directory %APPDATA%\tcACCESS.

2.1 Starting the Program

Start tcACCESS under Windows 3.1x or Windows NT 3.51 by clicking the tcACCESS symbol located in the tcACCESS group. For more current Windows versions (Windows 9x, Windows NT 4.0, Windows 2000, Windows XP and Vista), you can find the "tcACCESS" entry in the "Start menu" - "Programs" .

When you access the program, and prior to activating a mainframe session, the tcACCESS window displays.

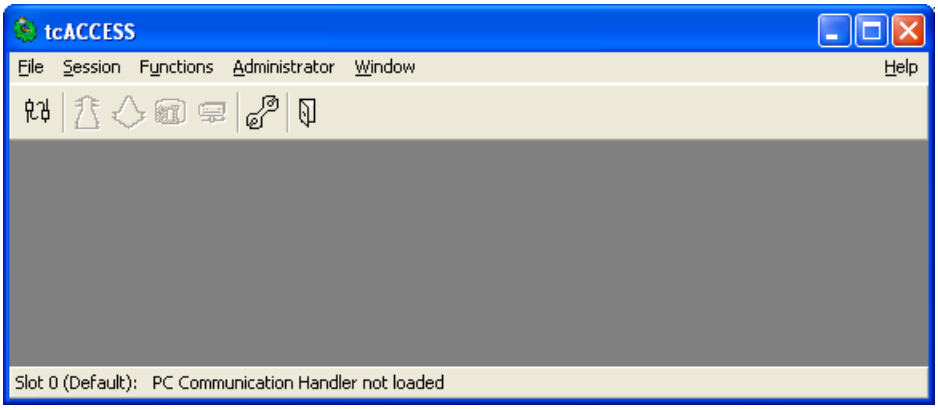


Fig. 15: tcACCESS Main Program (without any session)

Refer to the "Administrator" drop-down under "Host Connection Setup" for configuring the type of host link.
Once a mainframe session is allocated, or if it was allocated during the last program start, the tcACCESS window displays.

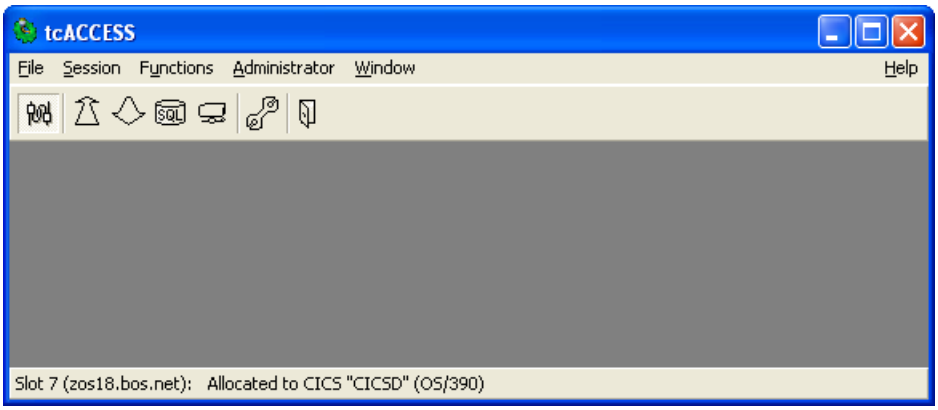


Fig. 16: tcACCESS Main Program (having an active session)

The window components are described below.

Main Menu

The items on the main menu are described in detail in the section Menu Structure*, page 33*.

Toolbar

Use the toolbar symbols to start frequently used tcACCESS functions directly. If these symbols are completely greyed (including the black components), they are not activated (e.g., when there is host session).

The tcACCESS program contains the following symbols:

SYMBOL	DESCRIPTION	REFERENCE
	Start a host session	(see Allocate Session, page *89*)
	Terminate an existing host session	(see Deallocate Session, page *92*)
	Send data to the host	(see PC => Host Transfer, page *97*)
	Download data from the host	(see Host => PC Transfer, page *127)*
	Execute SQL commands on the host	(seeHost SQL Query, page *156*)
	Administrate virtual disks	(see Virtual Disks, page *179*)
	User settings	(see Options, page *93)*
	Close the tcACCESS main program	(see Exit, page *89*)

Status bar

The status bar displays the current status of a host connection, such as the following:

STATUS MESSAGE	DESCRIPTION
Slot n (configuration) : PC Communication Handler not loaded	After starting tcACCESS for the first time, the PC Communication Handler is not active. When the connection is established, it remains loaded and the message displays *Detached*.
Slot n (configuration) : Detached	A PC Communication Handler is loaded but there is no host connection established.
Slot n (configuration) : 3270 Mode	A host connection has been established and 3270 commands can be transmitted to the host. When this status displays, there are no active sessions.
Slot n (configuration) : Allocated to VM/CMS	The user is signed on to the host and there is a session with VM.
Slot n (configuration) : Allocated to CICS *name* (OS/390)	The user is signed on to the host and there is a CICS session allocated under OS/390 to the named CICS.
Slot n (configuration) : Allocated to CICS *name* (VSE)	The user is signed on to the host and there is a CICS session allocated under VSE to the named CICS.
Slot n (configuration) : Allocated to tcACCESS/VTAM	The user is signed on to the host and there is a session with the tcACCESS system monitor.

2.2 Menu Structure

You may select some items on the menu only if there is an active mainframe session. These items are no longer displayed in grey.

This applies to the following menu items:

MENU ITEMS	REFERENCE
Deallocate Session	see page *92*
PC – Host Transfer	see page *97*.
Host – PC Transfer	see page *127*
Host SQL Query	see page *156*
Virtual Disks	see page *179*
tcACCESS Host Security	see page *206*

MENU ITEMS	REFERENCE
Administration	

To use the Administrator sub-menu items, (see *The Menu "Administrator"*, page *185*), the user must enter a valid password into the Administrator Logon, page *186).* Users may also use the *Use Administrator Signon* option.

You can deactivate menu items by using the Administrator, see section Tab Dialog "Menu customization".

2.2.1 The File Menu

2.2.1.1 Field Definition Import

To access host data using the SQL-Engine, or when processing a structured data transfer, you must define the data source structure.

A text file in .FTB format must be used for the tcACCESS SQL-Engine. The text file must reside in a defined directory of a virtual disk on the host (see also the *tcACCESS CICS Installation and tcACCESS VTAM Monitor installation*).
A text file in .PST format (structure file for the Host => PC Transfer and PC => Host Transfer dialogs) must reside on the PC to perform a structured data transfer. You can generate both format files using the Import of Field Definitions dialog.

tcACCESS currently supports the following formats for importing field definitions.

- ADABAS (Predict)
- Assembler
- COBOL/COBOL2
- CPG
- DATACOM/DB
- PL/I
- KDBS
- DL/I
- IDMS schema file

Due to the modular product structure, adding the appropriate import filter can support additional formats. If you require additional formats, contact your local B.O.S. representative.

The following table describes the buttons on each dialog:

MENU ITEMS	REFERENCE
< Back	Displays the preceding import step. Settings and/or selections may be changed.
Next > / Create	Allows the user to proceed to the next step (section) of the import process. **Next** displays until the last section of the import process is reached; then it changes to **Create**.
Next > / Finish	As long as the last section of an import process is not shown, this button has the inscription **Next**, otherwise it changes to **Finish**. It allows you to go to the next step (section) of processing or to complete it.
Cancel	Cancels the dialog without performing any operation.
Help	Displays online help for the open dialog.

2.2.1.1.1 Section: Source to use for importing field definitions

To import field definitions, you first specify the source containing the structure information that you want to convert. If there is an active host session, locate the input file on the mainframe. Otherwise, the Host file tab dialog is deactivated and you must define a local PC as the source.

Click the **Next** button to specify additional processing relating to the open tab dialog. If additional user input is required, a dialog displays. The components in the dialog may differ depending on the import module that is active (see Import Module Specific Dialogs, page *41)*.

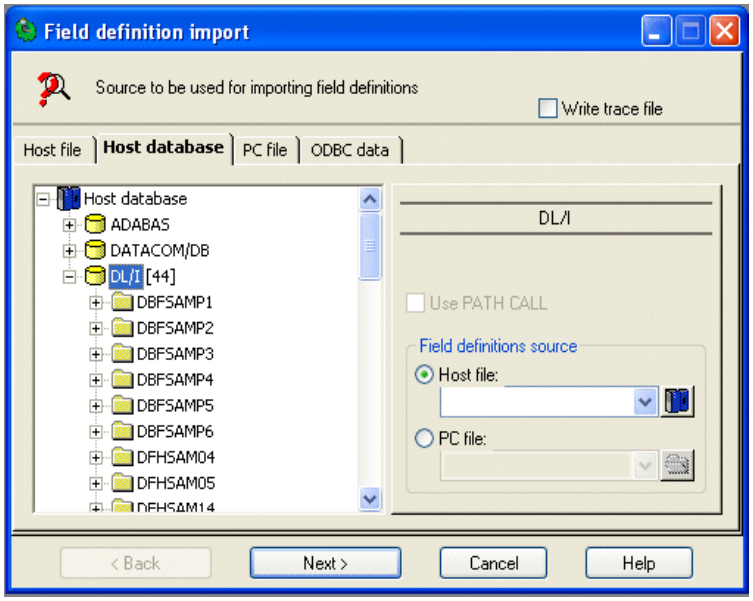


Fig. 17: Field definition import - Source to be used for importing field definitions

The dialog components are described below:

Write trace file

If you select this option, detailed information on the progress of the import process is logged in a temporary file.

Host file Tab Dialog

Tree view

Depending on the connected host operating system, the tree view contains the following file types:

OS/390 -	MVS PDS/PS
VSE -	POWER Queues, VSE-Library, VSE-Library (Catalog)
VM -	VM/CMS

When you select a file from the tree view, the name of the file displays in the HOST FILE field. Only the file name in this field is used for processing. To search for a file when using the MVS PDS/PS tree view in an OS/390 environment:

1. Enter the First Level Qualifier in the dialog window.

   This is the part of the name up to the first dot. The name is system-specific and is used to search for existing MVS datasets.

   Example: In order to find the SYSTEM.TEST.TMP file, enter the First Level Qualifier SYSTEM in the dialog box.

   If the read operation is successful, a second dialog box opens automatically.

2. Enter the First Level Qualifier for MVS PDS/PS in the dialog box.

   Notes: The MVS PDS/PS dataset may consist of several Members. When you open the dataset, all existing Members are detected and displayed. Each Member type in the tree view can be expanded to display additional data.

Similar processing applies to the "VSE-Library (Catalog)"-tree-view on VSE-systems. VSE-libraries can be displayed that do not have a LIBDEF-entry in the CICS-startup job (JCL). Click on the "+"-sign of the tree-view and a dialog box displays asking you for the name of the VSAM-file that contains the desired sublibraries. If the VSAM file is not part of the Master-Catalog the appropriate User Catalog must be defined. Click on "OK" and an additional tree-view opens that presents the selected VSAM library (i.e. "VSE-Library (VSE.BOS.LIBRARY)") and displays the sublibraries and members. Using this method multiple libraries can be displayed, each one with a tree-view of its own.

A different processing applies to libraries that are part of the LIBDEF-entry. These libraries will be automatically displayed.

If the VM/CMS File sub-tree is opened while the Shift key is pressed, a dialog box opens. Enter the filter conditions (e.g., * EXEC *). If no key is pressed when it is opened, it lists all available CMS minidisks (File mode, FM).

Host file

Define the host file that is to be used to import field definitions, using the following syntax:

FILE TYPE	SYNTAX / EXAMPLE	
MVS PDS	Syntax: Example:	A name separated by dots followed by a member name in brackets COBOL.COPYLIB(DEF1)
MVS PS	Syntax: Example:	A name separated by dots COBOL.COPYLIB1
VSE-Library	Syntax: Example:	A name separated by periods in the format of: *Library.Sublibrary.Member.Typ* (VSELIB) must be added to the name. BOS.TESTLIB.COPYBOOK.A(VSELIB)
VSE-Library Catalog	Syntax: Example:	A name separated by periods in the format of: *Library.Sublibrary.Member.Typ* (VSELIB) must be added to the name. TESTLIB.BIN.PHASE(VSELIB)
VM/CMS	Syntax: Example:	(CMS) must be added behind the CMS file description, which includes the file name, file type, and file mode (minidisk) each separated by blanks. COBOL COPYLIB B(CMS)
POWER Print Queue	Syntax:	LST(POWER) ,JN=job_name,JR=job_userid,CL=class,UI=from_node
POWER Punch Queue	Syntax:	PUN(POWER) ,JN=job_name,JR=job_userid,CL=class,UI=from_node
POWER Reader Queue	Syntax:	RDR(POWER),JN=job_name,JR=job_userid,CL=class,UI=from_node

If the specified host file is a MVS PDS/PS-type file, additional entry formats provide for Disposition for OS/390 read access and DISK NAME (VOLSER).

Special input fields for data-type "VSE-Library" or "VSE-Library (Catalog)":

Use VSAM-name

If the VSE-library has not been defined with a LIBDEF-definition, this check box must be activated and the name of the VSAM file must be defined in the text input field.

Catalog

If the file specified with "Use VSAM-name" is part of a User Catalog and not a Master Catalog, the name of the User Catalog must be defined.

Record format

The record format of the member can be defined. Possible selections are "(U)ndefined", "(F)ixed" and "(V)ariable". If the format is "Undefined" and "Variable" the record length must also be specified.

Logical record length

The logical record length of a library with record format "Undefined" or "Variable".

Special input fields for data-type "MVS PDS/PS":

Disposition for read access

This selection group is only available with the OS/390 host operating system. It allows you to define either the OLD or SHR mode for read access.

Disk name (VOLSER)

This field is only available with the MVS OS/390 host operating systems. An entry in this field allows you to define a disk name (Volume Serial) for non-catalogued MVS PDS/PS files.

Host database Tab Dialog

Tree view

The tree view contains entries for any existing ADABAS object and DATACOM/DB table. Its name, additional DB number parameters, and File number represent each ADABAS object. By clicking on the DL/I +-sign a dialog will be opened to define the desired PSB(VSE) or the PSBLIB and DBDLIB (OS/390).

Right-mouse clicking on the DATACOM/DB entry in the tree view will show a context menu that displays the menu entries "Display Sql-Names" and "Display Entity-Names". Selecting one of the entries will either display the Sql-Names or the Entity-Names of the DATACOM/DB tables in the tree view. Selecting option "Display Sql-Names" will display the SQL-name

followed by the entity name in brackets. If no SQL-name for a DATACOM/DB table exists, only the entity-name will be displayed.

ADABAS Parameter group - Object Name, DB Number, File Number, Password

The combination of the three parameters—OBJECT NAME, DB NUMBER, and FILE NUMBER—references the desired ADABAS object whose structure is analyzed and converted during the import process.

If a password is needed for the ADABAS-access it can be specified in the input field. In this case the password will also be needed for the display of the tree-view and must be specified before the tree-view is being displayed.

DATACOM Parameter group - Entity name, Status, Version, DATACOM name, and Database ID

The combination of the three parameters—Entity name, Status, and Version—references the selected DATACOM/DB table whose structure is analyzed and converted during the import process. Additional parameters, DATACOM name and Database ID, are only required for accessing the table content and appear as HOSTFILE= and DBID= within the created SQL-Engine table definition file.

Using the check-box "Display redefines" you can specify whether fields that redefine other fields should be processed during the import and should become part of the field definitions. Otherwise these fields will be ignored.

DL/I Parameter group – use PATH CALL, field definitions source

When selecting the tree view entry DL/I, input fields will be displayed on the right side of the dialog. For IMS/DB under OS/390 the PSBLIB and DBDLIB will be processed, to obtain the needed information for constructing the SSA. For field names, you can use field "Field definitions source" (PDS or PC file containing COBOL copy books, for example) . The obtained information from the PSB and DBD will be used to construct the SSA and will be automatically inserted into the SQL-Engine table definition.

Note: If the input file is a VSE Library member, the "VSAM-Qualifier" or "Catalog-Name" must be defined. They can be specified in the tree view. The tree view will be displayed when you select the "Browse" button in panel "Field definition import".

The selection of segments is done in the tree view. The segments will be displayed in their hierarchical order. You must select a PCB. After the selection, all available segments will be displayed.

Option "Use PATH CALL" is available if the corresponding entries in the PCB have been found.

PC file Tab Dialog

PC file name

The defined PC file is used to enter the import field definitions. When you click the **Search** button, a dialog box displays and allows you to browse through the directory.

Tab "ODBC data"

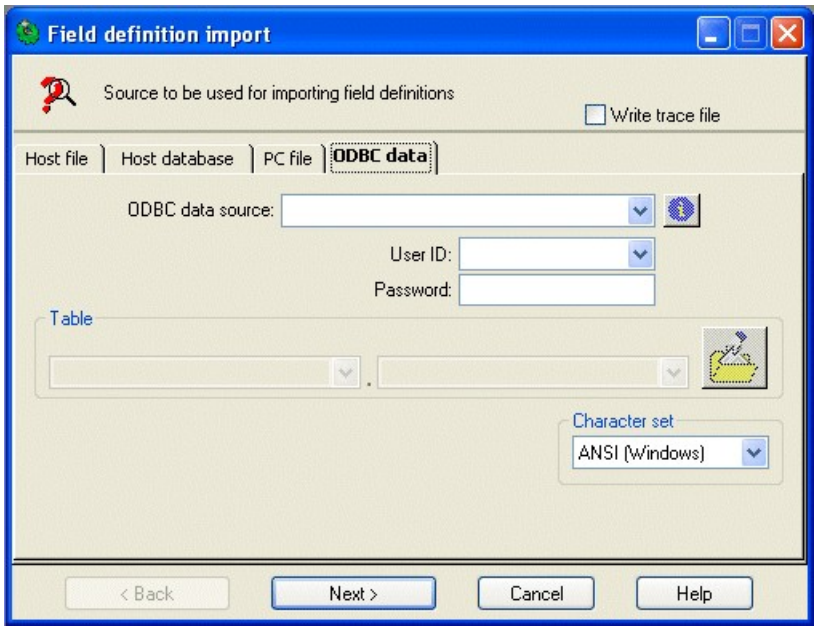


Fig. 18: Import of field definitions - Tab "ODBC data"

ODBC data source

A list of all data sources installed on the PC will be displayed. To add a new data source you should use the corresponding dialog in the Windows control panel.

Clicking on the Info-button will display additional information about the selected data-source. This includes the description, the parameter and the conversion between the tcACCESS data-types and the internal data-types of the ODBC data source.

User ID

Specify a user ID if the data source requires one.

Password

Specify a password if the data source requires one.

Table

A table name has either the format "**CREATOR.TABLE**" (i.e.: "MS SQL Server") or only "**TABLE**" (i.e.: "MS Access"). If the selected ODBC data source does not use a "Creator" ("Owner" etc.) the left input field will not be active. Clicking on the button on the right hand side will display all tables and views provided by the ODBC data source. The entry can be selected or can be specified manually into the input field.

Character set

Some ODBC-driver do not use the Windows character set (ANSI) but use the DOS character set (OEM). This can be recognized with special characters ("ü", "ä", "ß", ...). If you experience translation errors you should change the specified character set.

2.2.1.1.2 Import Module Specific Dialogs

Depending on the import module that analyzes the data source, additional dialogs display to enter more detailed specifications as required.

2.2.1.1.2.1 ADABAS Predict

Under certain conditions, the tcACCESS import filter for ADABAS Predict requires user input since the system cannot automatically determine some information. The input values are saved in the tcACCESS.INI file after you click the **OK** button. The input values become predefined settings, and are used when the dialog is re-opened.

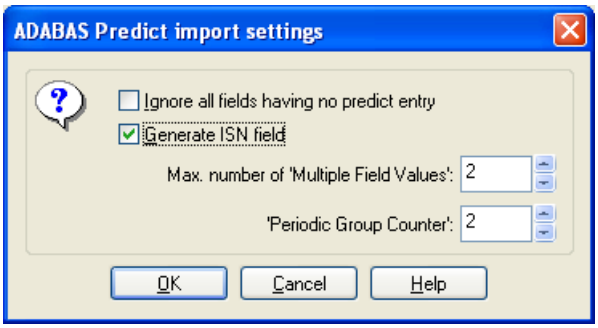


Fig. 19: Field definition import - ADABAS Predict import settings

The following table describes the dialog components:

DIALOG COMPONENT	DESCRIPTION
Ignore all fields having no predict entry	If this option is selected, only definitions having a predict name are imported. Otherwise, all definitions are used and, if applicable, a field name is generated automatically (e.g., ADABAS_AD).

DIALOG COMPONENT	DESCRIPTION
Generation of an ISN field	If this option is active, a full word field with a name of ADABAS_ISN will be generated at the end of the field list. This field allows access to the internal ADABAS record number.

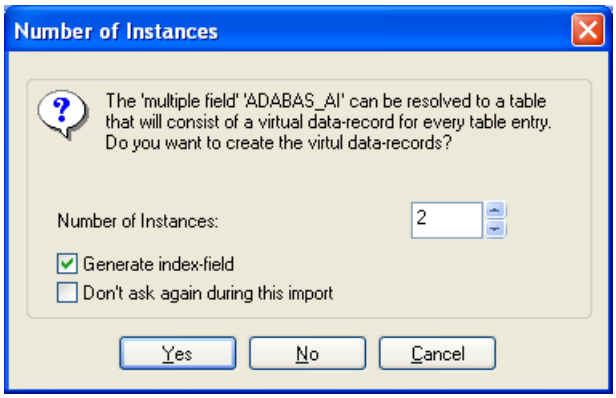


Fig. 20: Field definition import - ADABAS Predict import settings

The dialog components have the following meaning:

Number of Instances

Specifies the maximum number of entries in the table. The value will be taken from PREDICT. If no PREDICT is available, the stored values will be displayed and can be changed.

Generate Index-field

For each table entry a data-record will be built. In this data-record the position of the table entry can be included as a half word.

Yes/No

Specifies whether the table will be resolved into individual data-records (virtual table) or whether the table will become part of the data-record as a "flat" structure.

Don't ask again during this import

This dialog will be displayed once for every table. If you do not want the dialog to be displayed again, activate this control box. All tables will be generated with the PREDICT values. Entries not found in PREDICT will have the last specified value.

2.2.1.1.2.2 COBOL, PL/I and KSDS

If a COBOL Copybook contains multiple 01 levels, a dialogue is displayed to select a segment level.
The selected values will be saved into "tcaccess.ini".

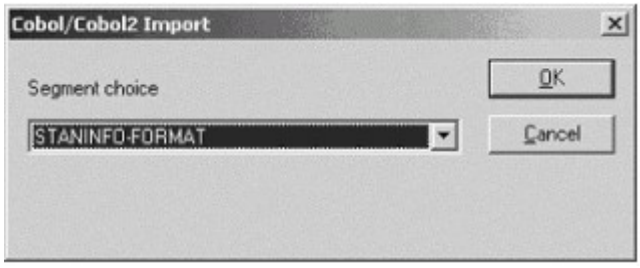


Fig. 21: Import of field definitions – segment selection

Segmentname

Select the segment, 01 level or component of the imported file.

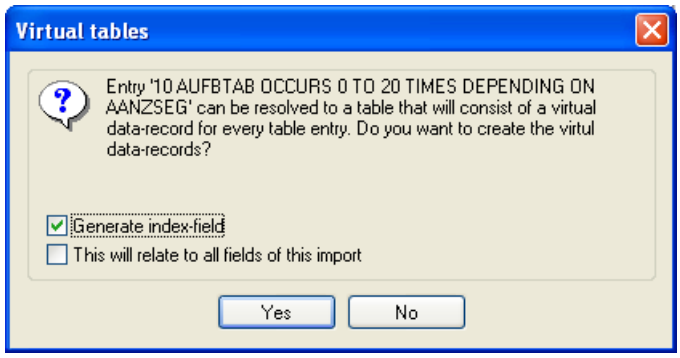


Fig. 22: Import of field definitions – segment selection

The dialog components have the following meaning:

Yes/No

Specifies whether the table will be resolved into individual data-records (virtual table) or whether the table will become part of the data-record as a "flat" structure.

Generate Index-field

For each table entry a data-record will be built. In this data-record the position of the table entry can be included as a half word.

This will relate to all fields of this import

This dialog will be displayed once for every table. If you do not want the dialog to be displayed again, activate this control box.

Section: Result of import processing

This dialog displays information and error messages relating to affected imported field definitions. If the process is unsuccessful, the **Next>** button is not available. In this case, it is possible to use the **< Back>** button to go back one section and to start the import process again.

However, if tcACCESS can process the input file, this section shows the file type (e.g., Cobol/Cobol2 Copybook) and the number of successfully imported field definitions.

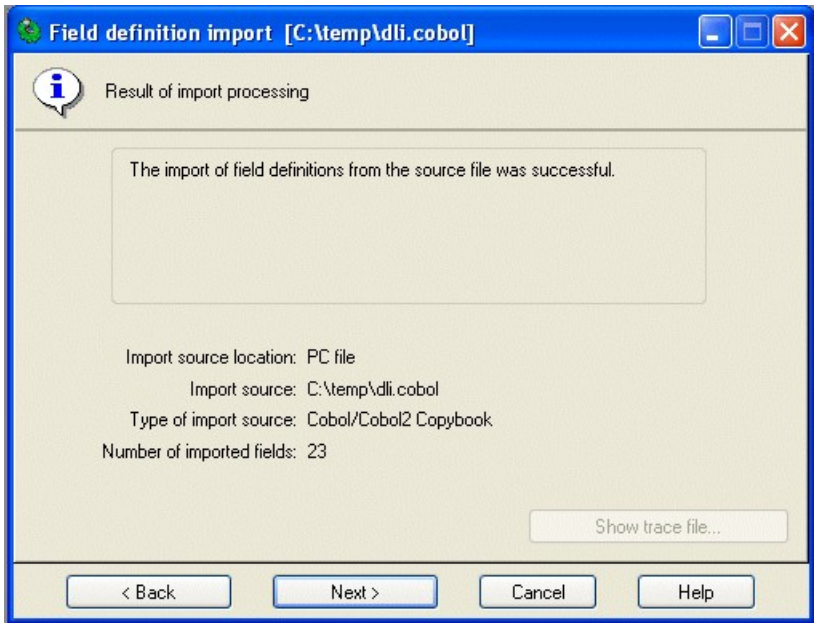


Fig. 23: Field definition import - Result of import processing

The dialog components are described below:

Show trace file

If the WRITE TRACE FILE option was activated in on the Field Definition Import dialog, you can select this button to examine the text file. The selected program displays in the PROGRAM FOR TEXT DISPLAY OPTIONS field (see Options, page *93*).

2.2.1.1.3 Section: Options for importing field definitions

When the field definitions import successfully, this section allows you to specify the settings for additional processing.

During a conversion, it is possible to switch between this and the next page of the wizard. Any changes immediately update when you click the **Next >** button.

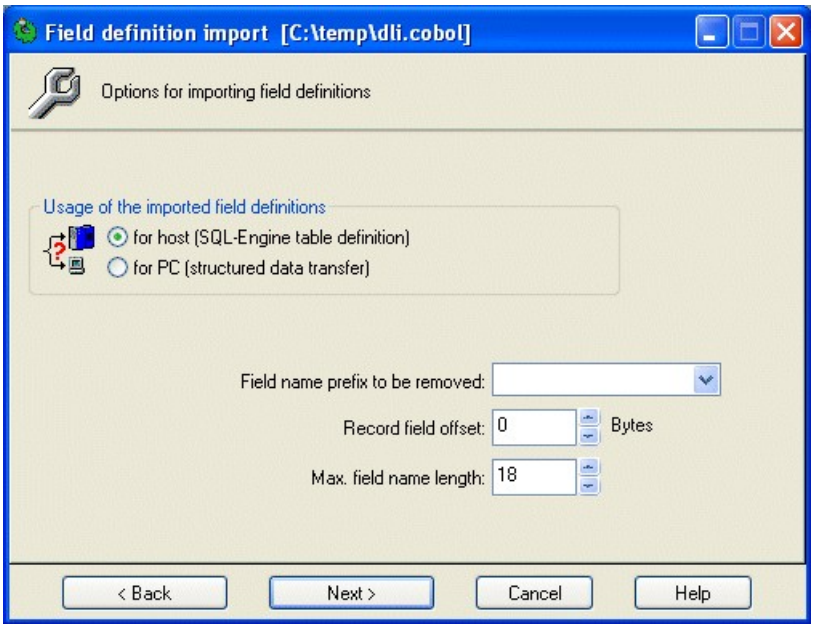


Fig. 24: Field definition import - Options for importing field definitions

The following table describes the dialog components and the available options:

TYPE	DESCRIPTION
for host (SQL-Engine table definition)	The file is used as an SQL-Engine table definition on the host. This file must be saved on the virtual disk.
for PC (structured data transfer)	The file is used as a definition file for the fields when a structured file transfer is performed. This file must be saved on the local PC. It is saved as a structure file (*.pst) and can be used by applying the Load parameters menu item for a Host → PC transfer or a PC → Host transfer. In the following sections of this manual this type of definition is called PST format.

Field name prefix to be removed

This field allows you to define a prefix to be removed from the beginning of all imported field names. This allows you to remove identical parts of a name (e.g., the file name) in order to generate descriptive names.

Note: All field names are shortened to the length defined in MAX. FIELD NAME LENGTH and then made unique. A specified prefix is removed first to make the resulting field name identifiable. All hyphens within names are replaced by underscores if they are used as host SQL-Engine table definition files.

Record field offset

If this numeric field contains a value other than zero, the offset of all fields is shifted by the value entered. Keys or data fields at the beginning can be skipped if the copybook includes only part of the complete record definition.

Max. field name length

While generating an SQL-Engine table definition file, the maximum length of the field names is determined by applying the MAXFLEN generation parameter on the host (default = 18). This field must contain the same value or a shorter length specification in relation to the host generation. However, if the PST format for a structured file transfer is generated, the maximum length of the field names is 20 characters.

The field names are shortened (after the prefix is removed) to the defined number of positions. Appending *_nnn* makes them unique. The only exception is fields with the name FILLER, which are not renamed since they are handled in a special manner (ignored) by the SQL-Engine.

2.2.1.1.4 Section: Check/edit imported field definitions

This process displays all fields created during the import process.

Since hyphens within field names of an SQL-Engine table definition are not permitted, they are replaced by underscores for system use. Additionally, all field names are shortened to the defined number of positions and made unique.

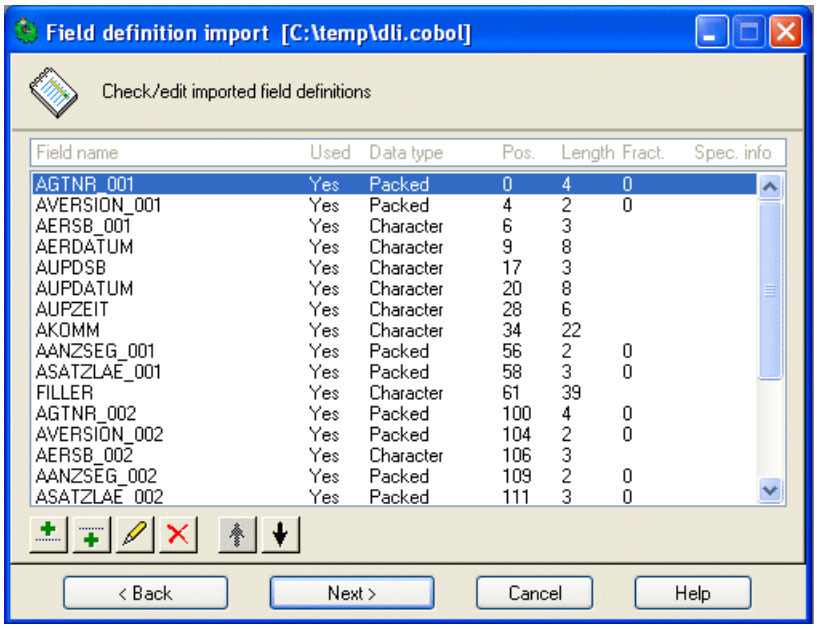


Fig. 25: Field definition import - Check/edit imported field definitions (SQL-Engine Format)

For a structured data transfer, hyphens within field names are not replaced, since the PST format does not have the same limitation as the SQL-Engine table definition. The field names are shortened to the specified number of positions and made unique.

Additional information about processing fields in .PST format can be found in the section *Edit field definition,* on page *148*.

format)

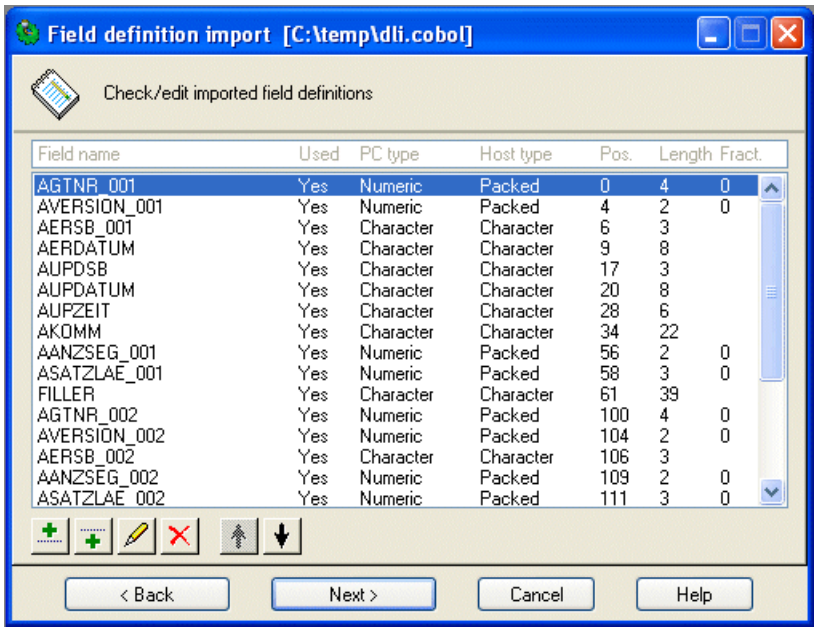


Fig. 26: Field definition import - Check/edit imported field definitions (structured data transfer

The dialog components are described below:

List of field definitions

This list contains all defined fields with their most important criteria. To edit a field, double-click on an entry or click the **Edit** button. Select the **Delete** key to remove the selected list entry. Press the space key to activate or deactivate the Used for query option.

The list of field definitions (in SQL-Engine and structured data transfer format) consists of the following columns:

COLUMN NAME	DESCRIPTION
Field name	The table column name, which is used for SQL queries with the SQL-Engine.
Used	When the parameter is set to YES, this field displays and is used for SQL queries.
Data type / PC type	Target data type of the field on the Host / PC.
Pos.	Beginning of the field within a record. A subsequent '*' character shows that the position is calculated automatically based on the preceding field's position and length. The first byte of a record has position 0.
Length	The field length in bytes.
Fract.	Number of positions behind the decimal point for the host data types Zoned, Packed, Halfword, and Fullword.
Spec. info	Supplementary information depending on the type of data source, e.g., the corresponding ADABAS name.

Insert

Clicking this button opens a window that allows you to create a valid field definition. When you create a new field definition, the highlighted field definition moves one position down. (See the following section Edit Field Definition (SQL-Engine format) page *148*).

Append

Clicking this button opens a window that allows you to create a valid field definition. The new field definition appends to the highlighted field definition. (See the following section Edit Field Definition (SQL-Engine format) , page *148*).

Edit

Clicking this button opens a window that allows you to process the highlighted field definition (See the following section Edit Field Definition (SQL-Engine format) or, page *148*).

Delete

Removes the highlighted field definition from the list.

Move up

Moves the highlighted field definition one position upward.

Move down

Moves the highlighted field definition one position downward.

2.2.1.1.5 Edit Field Definition (SQL-Engine format)

This dialog allows you to verify and, if necessary, modify an SQL-Engine format field definition.

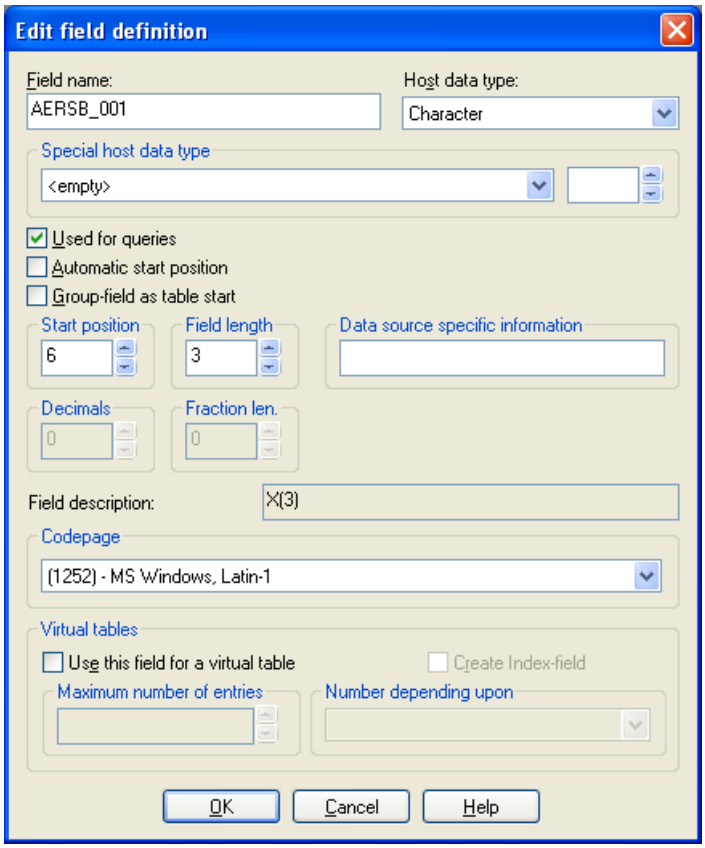


Fig. 27: Edit Field Definition (SQL-Engine Format)

The dialog components are described below:

Field name

A unique name to identify a defined field. The maximum length of SQL-Engine format field names is determined by the MAXFLEN generation parameter on the host (default = 18). The field must not contain any blanks. If the original field name contains hyphens, they are replaced by underscores. The SQL-Engine ignores fields named FILLER.

Host data type

This defines the type of field. The following types are available:

COLUMN NAME	DESCRIPTION
Character	EBCDIC character string (COBOL PIC X(xx))
VARCHAR	EBCDIC character string with length indicator. (2 Bytes in front of data). Maximum length is 32765 characters (32767 including length indicator).
VARCHAR without length indicator	EBCDIC character string. Maximum length can be 32767 bytes.
Hexadecimal	Data is displayed in hexadecimal form.

<table>
<tr><th>COLUMN NAME</th><th>DESCRIPTION</th></tr>
<tr><td>Zoned</td><td>Numeric value in zoned format (variable length)
(COBOL PIC S9(xx))</td></tr>
<tr><td>Packed</td><td>Numeric value in packed format (variable length)
(COBOL PIC 9(xx) COMP-3)</td></tr>
<tr><td>Halfword</td><td>Numeric value, 2 bytes
(COBOL PIC 9(4) COMP)
A 1-byte long and unsigned integer may be used by providing the special data type 2.</td></tr>
<tr><td>Fullword</td><td>Numeric value, 4 bytes
(COBOL PIC 9(8) COMP)
A 3 byte long and unsigned integer may be used by providing the special data type 3.</td></tr>
<tr><td>Date</td><td>Date field
The following date formats may be selected by using a Special host data type:</td></tr>
<tr><td></td><td>
<table>
<tr><th>Number</th><th>Host data type</th><th>Length</th><th>Format</th></tr>
<tr><td>0 (Standard)</td><td>Character</td><td>10 bytes</td><td>YYYY-MM-DD</td></tr>
<tr><td>1</td><td>Packed</td><td>3 bytes</td><td>YYDDD (half bytes)</td></tr>
<tr><td>50</td><td>Binary</td><td>4 bytes</td><td>YYMD (bytes)</td></tr>
<tr><td>51</td><td>Character</td><td>6 bytes</td><td>YYMMDD</td></tr>
<tr><td>52</td><td>Binary</td><td>2 bytes</td><td>YYYYYYYMMMM DDDDD (bits)</td></tr>
<tr><td>54</td><td>Character</td><td>10 bytes</td><td>YYYY.MM.DD</td></tr>
<tr><td>55</td><td>Character</td><td>8 bytes</td><td>YYYYMMDD</td></tr>
<tr><td>56</td><td>Fullword</td><td>4 bytes</td><td>DDDD (bytes)</td></tr>
<tr><td>57</td><td>Packed</td><td>4 bytes</td><td>ADABAS date format</td></tr>
</table>
</td></tr>
</table>

Timestamp	Date and time (Format: YYYY-MM-DD-HH.MM.SS.TTTTT) A 7-byte long ADABAS timestamp format may be used by providing the special data type 80.
Time	Time field (Format: 'HH-MM-SS') A 7-byte long ADABAS time format may be used by

	providing the special data type 90.

Special host data type

Use this field to select a special data type (see the description of the host data types above). When you select User defined data type the input field beside it is activated and the number of selected data types may be entered (128 - 255).

It is possible to extend the displayed list. In order to do this, you should create a configuration file with a name of "userfld.ini" in the configuration directory (default "C:\Program Files\tcACCESS\Config"). This file then should contain a section for any supported language (i.e. "[German]"). In this section you can specify multiple user fields in the following format: "NR_DESCRIPTION". Using the "[Rules]" section you can specify the corresponding host data type and the length of the field within brackets.

Example of file "userfld.ini":

```
[German]
131=Spezielles Datum
132=Spezielle Zeichenkette mit fester Länge 25

[English]
131=Special Date
132=Special string having fixed length 25

[Rules]
131=DATE
132=CHAR(25)
```

Available language sections:

"**[American]**"	USA English
"**[English]**"	English
"**[French]**"	French
"**[German]**"	German
"**[Italian]**"	Italian

Available host data types in "[Rules]"

"**CHAR**"	Character
"V**ARCHAR**"	Varchar
"**VARCHAR2**"	Varchar without length indicator
"**HEX**"	Hexadecimal
"**ZONED**"	Zoned
"**PACKED**"	Packed
"**HALFWORD**"	Halfword
"**FULLWORD**"	Fullword
"**DATE**"	Date
"**TIME**"	Time
"**TIMESTAMP**"	Timestamp

For further information refer to chapter "Exit program for user specific data fields" (Manual "Host Server") and chapter "Conversion of date formats" (Manual "tcACCESS Customization") for a description of macro TCADCONV.

Used for queries

Lets you determine whether the appropriate field should be used or ignored for SQL queries with the SQL-Engine.

Automatic start position

If this option is selected, tcACCESS automatically calculates the start position of a field based on the position and length of the preceding field. This option is inactive by default for fields created by the import function.

Group-field as table start

Specifies whether the field should be used as a group field for a table. Insert this field when a virtual table consists of more than one field.

Start position

This field is only available if the Automatic Start position option has been deactivated. If so, you can use this field to manually enter the field offset within the host record.

Field length

This field allows you to define or determine the field length within the host record. Depending on the selected host data type, the following rules apply:

FIELD NAME	DESCRIPTION
Character	The maximum length is 255
Zoned	Uneditable. Content is calculated based of the value entered in the *Decimal pos*. field.
Packed	Uneditable. Content is calculated from the value entered in the DECIMALS field.
Halfword	Uneditable. It is always 2 bytes long.
Fullword	Uneditable.. It is always 4 bytes long.
Floating point	Uneditable. The length for this type of field is always 8 bytes.
Decimals	This value is to determine the number of decimals for the host data types Zoned and Packed. With all other types of data, this field is not considered.
Fraction len.	This value is to determine the number of positions behind the decimal point for the host data types Zoned, Packed, Halfword and Fullword.
Data source specific information	This field allows you to define options that are added at the end of each entry within the SQL-Engine table definition file (e.g., the corresponding ADABAS name).
Field description	This field is populated during the import and generally shows the original line of the field description (for COBOL e.g., the PIC field).
Virtual tables	This group contains all settings for virtual tables. As an example, these tables will be generated as a result of an import of a COBOL copy-book containing 'OCCURS' or 'OCCURS DEPENDING ON' clauses.
Use this field for a virtual table	If the ceck box is active, this field represents a virtual table.
Generate index-field	For every table entry a data-record will be constructed. The position in the table will be stored in this field as a half word.
Maximum number of entries	This input field contains the maximum numbe rof entries, that can exist in a data-record. The entry must match the definitions of the original data-record. It is identical to the numbe rof OCCURS in COBOL.
Number depending upon	Using this input field an additional numeric field can be specified that contains the counter of the actually used table entries in the data-reocd. The field is identical to the 'DEPENDING ON' clause in COBOL.
CCSID	Select the CCSID from the list that should be used for this field. This value will be used when the "Global Language Pack" is installed. The selection applies to character fields.

OK

When you click OK, all specifications for the field are verified. If there is an error, a message displays. If all values are accepted, the processed field returns to the caller and the dialog closes.

2.2.1.1.6 Section: Additional parameters for SQL-Engine table definition on host

This dialog appears only when the SQL-Engine format is selected. It allows you to define additional parameters, which may differ depending on the type of data source.

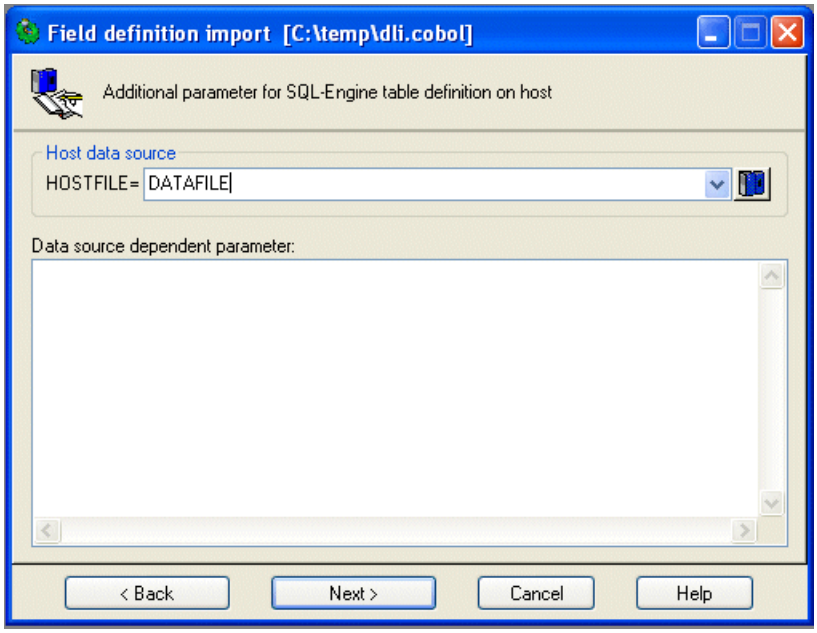


Fig. 28: Field definition import - Additional parameters for SQL-Engine table definition file on host

The dialog components are described below:

Host data source

The entry HOSTFILE= within an SQL-Engine table definition determines the host data source for queries using this SQL-Engine definition file. If the button next to it is selected, you can select the data source name from a tree view when a host session is active.

Data source dependent parameters

The SQL-Engine table definition file generates the data in this field without any changes. Therefore, it is possible to define any data source-related parameters as required. This is required, e.g., for ADABAS, DATACOM, DLI. A detailed description can be found in chapter 2.4.1 Table definition of the Host Server manual.

2.2.1.1.7 Section: Target to receive the imported field definitions

The result of the conversion is saved in the target file. If a host connection is active and the usage type in the SQL-Engine table definition file is defined, you can save the file on the virtual disk. It is also possible to save the target file on a PC disk drive.

If you are connected with host tcACCESS version 3.01, tcACCESS 3.00 PTF 20 (CICS), or PTF 16 (tcACCESS/VTAM)), the tab dialog Host SQL-Engine definition file displays. A tree view displays, from which you can select a file on the virtual disk for processing.
If the destination file is on the local PC, enter the name of the file in the Complete SQL-Engine table name field. You can view a list of the files used last by accessing the drop-down menu next to the field. Clicking the **Search** button displays a dialog, which allows you to scroll through the directory.

Click the **Create** button. The target file is written and the dialog closes. The file specification on the tab dialog is used for processing. The system prompts for another import process if the target file was successfully created.

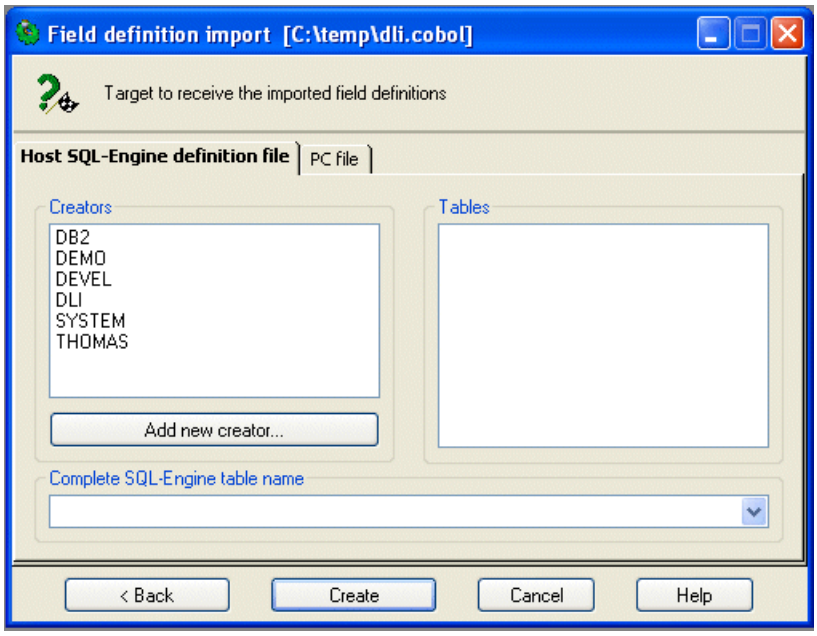


Fig. 29: Field definition import - Target to receive the imported field definitions

The dialog components are described below:

Host SQL-Engine definition file Tab Dialog

Creators

The list contains the creator names of all SQL-Engine defined tables. The entries are sub-directories on the SQL-Engine virtual disk. By selecting a creator, all table names belonging to it are retrieved and displayed in the Tables window.

Add new creator

Click this button to insert a new creator. It then provides the required creator name. A new sub-directory on the SQL-Engine virtual disk is created based on the name that you enter.

Tables

All table names associated with the selected creator displays in this window. By selecting an entry on this list an existing table definition is replaced. Each table belongs to an SQL-Engine table definition file (extension .FTB or .STP) within the creator's sub-directory on the SQL-Engine virtual disk.

Complete SQL-Engine table name

You must enter the complete name of the selected destination table in this field. The Creators and Tables lists provide a reference for the user. The syntax format of an SQL-Engine table name is CREATOR.TABLE.

PC file Tab Dialog

PC file name

The SQL-Engine table definition file (extension: .FTB) defined in this field is used to write the resulting field definitions. When you click the **Search** button, a dialog displays and allows you to scroll through the directory.

2.2.1.2 Import of Stored Procedure-definitions

When using a Stored Procedure a SQL statement can be used to start a host-program. As an example, this program can examine data, perform calculations etc. and can pass the result to the caller. A customization parameter specifies the directory on the Virtual Disk where the Stored Procedure definitions are kept. They have an extension of ".STP" (refer to manual "tcACCESS Customization parameter").

To access a Stored Procedure on the host using the SQL-Engine the data area of the host program must be defined.

Use dialog "Import of Stored Procedure-definitions" to generate the necessary definitions.

The following import formats are currently supported:

- **ADABAS (Predict)**
- **Assembler**
- **COBOL/COBOL2**
- **CPG**
- **DATACOM/DB**
- **PL/I**
- **KDBS**
- **DL/I**

- **IDMS (Schema-file**

The following table describes the buttons on each dialog:

MENU ITEMS	REFERENCE
< Back	Displays the preceding import step. Settings and/or selections may be changed.
Next > / Create	Allows the user to proceed to the next step (section) of the import process. **Next** displays until the last section of the import process is reached; then it changes to **Create**.
Next > / Finish	As long as the last section of an import process is not shown, this button has the inscription **Next**, otherwise it changes to **Finish**. It allows you to go to the next step (section) of processing or to complete it.
Cancel	Cancels the dialog without performing any operation.
Help	Displays online help for the open dialog.

2.2.1.2.1 Section ”Source to be used for importing Stored Procedure-definitions"

The first step during the import of Stored Procedure-definitions is the specification of a source that contains the structure information that must be converted. Ifa connection to the mainframe has been established at the time of the import, the source can also be on the mainframe. If no connection is established tab "Host file" is not active and a PC file must be selected as source.

The tab will be used as input that is displayed in the foreground when the ”Next >” button is pressed. If additional input is expected by the user a dialog box will be displayed whose appearance depends upon the currently used import module. Additional information concerning this step can be found in chapter Section: Source to use for importing field definitions on page 35.

2.2.1.2.2 Import-Module specific dialogs

Depending upon the import module used dialogs can be different. They are described in chapter Import Module Specific Dialogs on page 41.

2.2.1.2.3 Section "Result of the import process"

This dialöog displays information and error messages concerning the completed import of field definitions. If the import process has not been successful the ”Next >” button is not active. The ”Back <” button can be used to go back to the previous section and start the import process again.

If the import module has been able to process the data, this section displays the type of the file (i.e. "Cobol/Cobol2 Copybook") and the number of successfully imported field definitions.

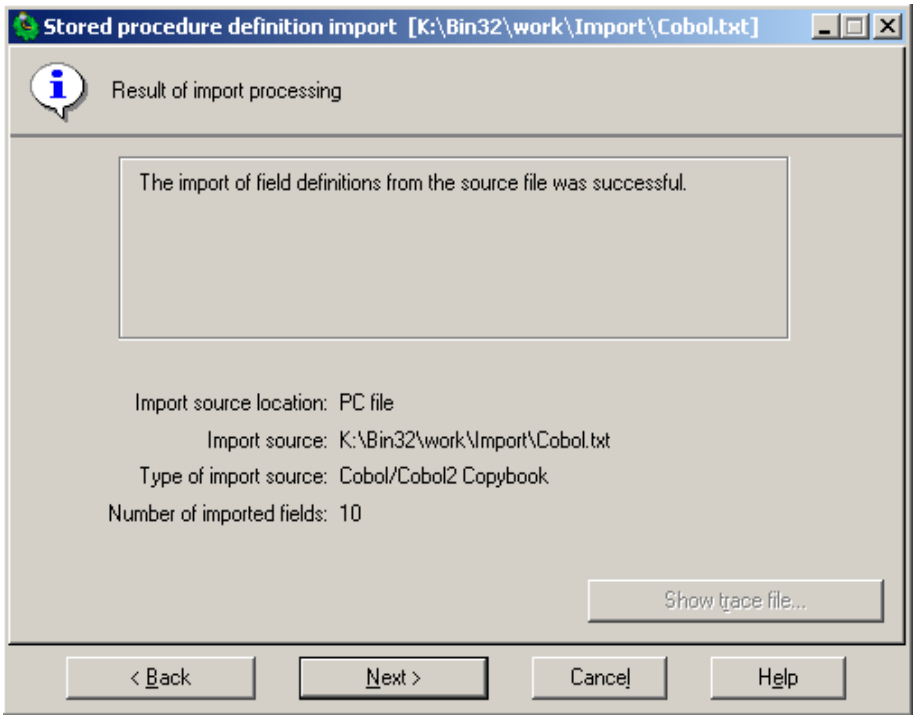


Fig. 30: Import of Stored Procedure definitions – Result of import processing

The dialog components have the following meaning:

Show trace file

If the trace file has been activated in the previous section of the import dialog, this file can be viewed here. The program defined in the options will be used to display the trace information (refer to Options, page 93).

2.2.1.2.4 Section "Options for the import of field definitions"

After the successful import of Stored Procedure definitions settings for the further processing can be defined in this section.

It is possible to switch between the next page of the wizard and this one during a conversion process. If changes have been performed they will be applied when using button ”Next >”.

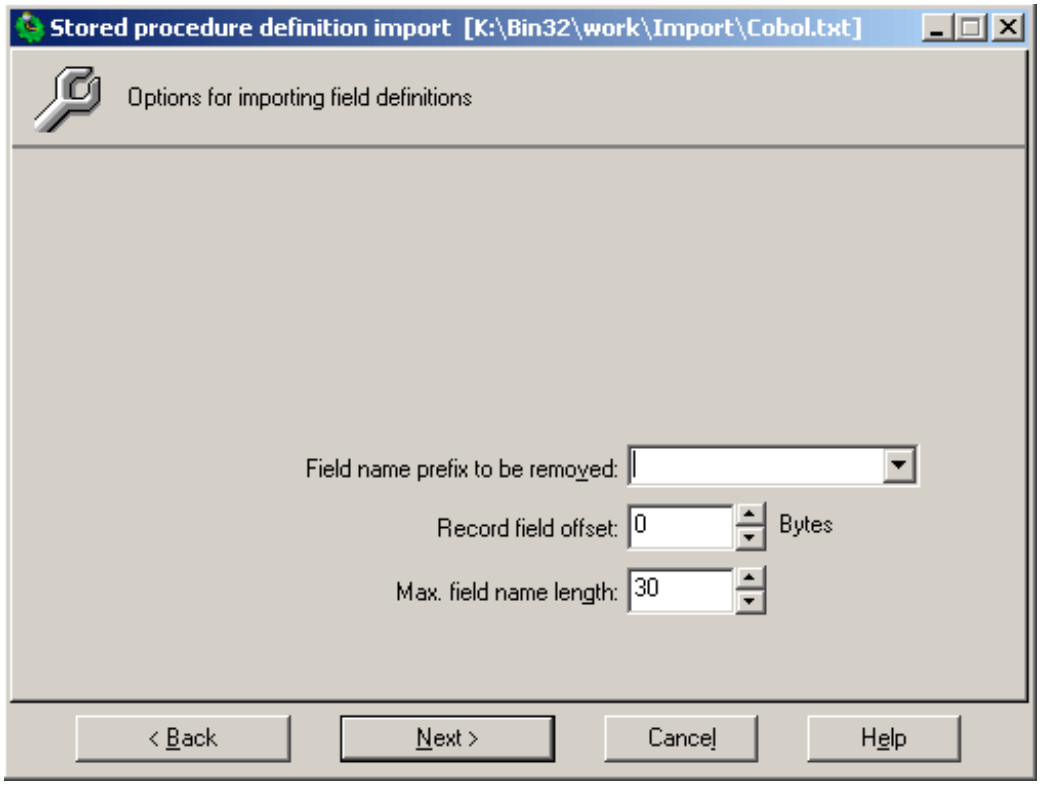


Fig. 31: Import of Stored Procedure definitions - Optionen for the import of field definitions

The dialog components have the following meaning:

Field name prefix to be removed

Using thios text box it is possible to define a prefix that should be removed from the beginning of all imported field names. This allows you to remove identical parts of a name (e.g., the file name) in order to generate descriptive names.

Note: All field names are shortened to the length defined in MAX. FIELD NAME LENGTH and then made unique. A specified prefix is removed first to make the resulting field name identifiable. All hyphens within names are replaced by underscores if they are used as host SQL-Engine table definition files.

Record field offset

If this numeric field contains a value other than zero, the offset of all fields is shifted by the value entered. Keys or data fields at the beginning can be skipped if the copybook includes only part of the complete record definition.

Max. field name length

While generating an SQL-Engine table definition file, the maximum length of the field names is determined by applying the MAXFLEN generation parameter on the host (default = 18). This field must contain the same value or a shorter length specification in relation to the host generation. However, if the PST format for a structured file transfer is generated, the maximum length of the field names is 20 characters.

The field names are shortened (after the prefix is removed) to the defined number of positions. Appending *_nnn* makes them unique. The only exception is fields with the name FILLER, which are not renamed since they are handled in a special manner (ignored) by the SQL-Engine.

2.2.1.2.5 Section "Check/Edit imported field definitions"

This process displays all fields created during the import process.

Since hyphens within field names of an SQL-Engine table definition are not permitted, they are replaced by underscores. Additionally, all field names are shortened to the defined number of positions and made unique.

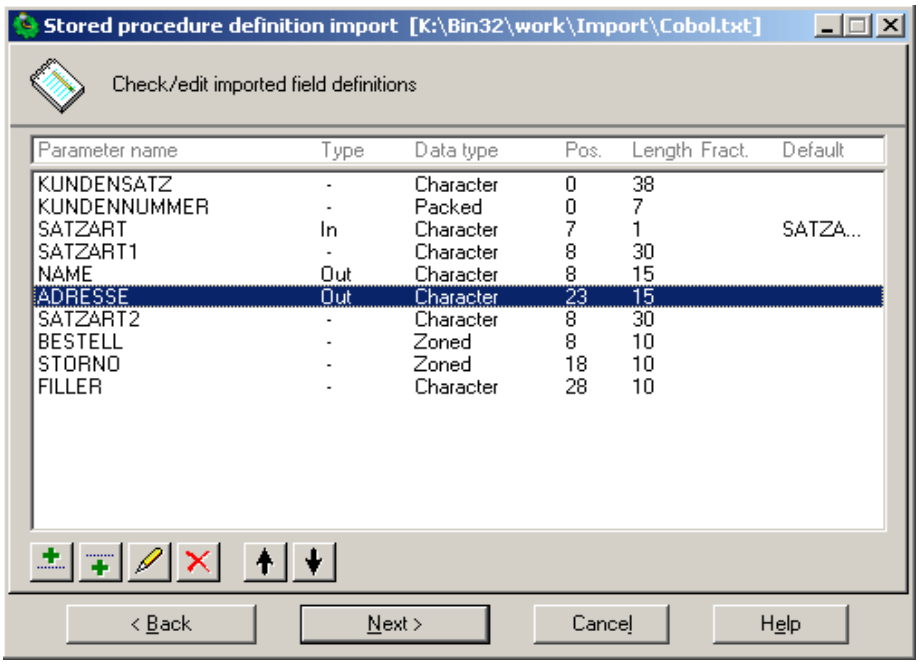


Fig. 32: Import of Stored Procedure definitions - Check/Edit imported field definitions

The dialog components have the following meaning:

List of field definitions

This list contains all defined fields with their most important criteria. To edit a field, double-click on an entry or click the **Edit** button. Select the **Delete** key to remove the selected list entry.

The list of field definitions consists of the following columns:

COLUMN NAME	DESCRIPTION
Paramete name	The name of the parameter when calling the Stored Procedure using the SQL-Engine.
Type	"IN" for input parameter, "OUT" for output parameter, "IN/OUT" for parameter used for input and output. "-" indicates that the parameter is neither used for input nor for output.
Data type	TData type of the field on the Host.
Pos.	Beginning of the field within a record. A subsequent '*' character shows that the position is calculated automatically based on the preceding field's position and length. The first byte of a record has position 0.
Length	The field length in bytes.
Fract.	Number of positions behind the decimal point for the host data types Zoned, Packed, Halfword, and Fullword.
Default	Default value for this field.

Meaning of the editing buttons:

 **Insert a field in front of the selected entry.**

 **Append a field after the selected entry**

 **Edit the selected entry**

 **Delete the selected entry**

 **Moves the selected entry one position up.**

 **Moves the selected entry one position down.**

The dialog displayed for the first three buttons is explained in the next chapter.

2.2.1.2.6 Edit field definition

A field definition of a Store Procedure can be maintained with this dialog.

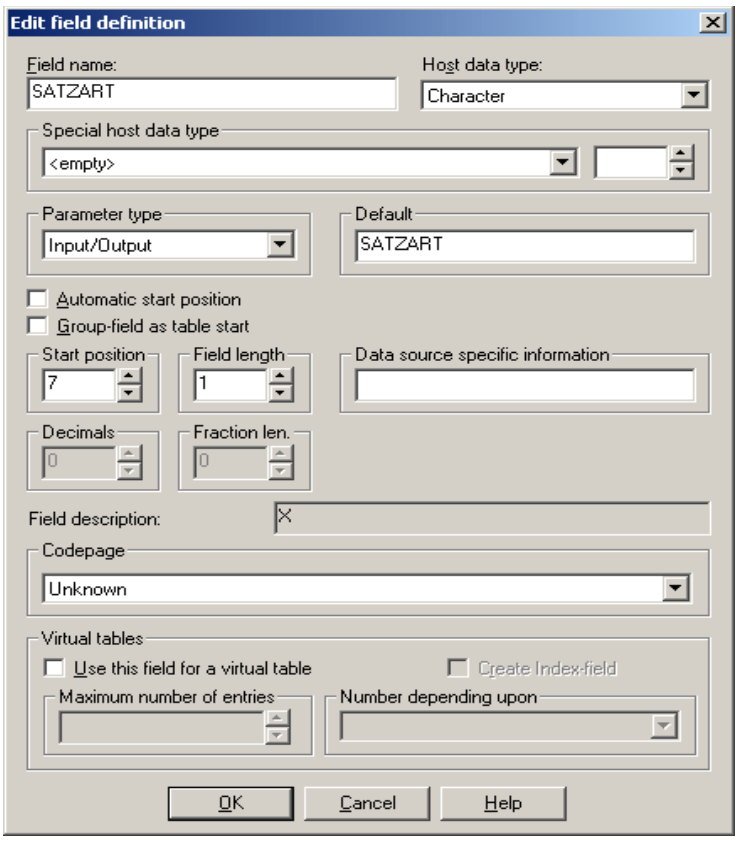


Fig. 33: Edit a feld definition for a Stored Procedure

The dialog components have the following meaning:

Parameter type

Specifies whether the parameter is of type INPUT, OUTPUT or INPUT/OUTPUT.

Default

A default value can be defined for output parameter (i.e. A character string or a value for a numeric field). For input parameter this field normally contains the parameter name.

Remaining dialog components

A detailled description of the other componetns can be found in chapter Edit Field Definition (SQL-Engine format) on page 48 .

OK

Clicking OK verfies all specifications made for the field. If there is an error, a message displays. If all values are accepted, the processed field returns to the caller and the dialog closes.

Cancel

Terminates the dialog and no action is performed.

Help

Display of the online help for this dialog.

2.2.1.2.7 Section "Resultset der Stored Procedure"

This dialog can be used to define additional resultsets, enabling the Stored procedure to also create these in addtion to the output parameter.

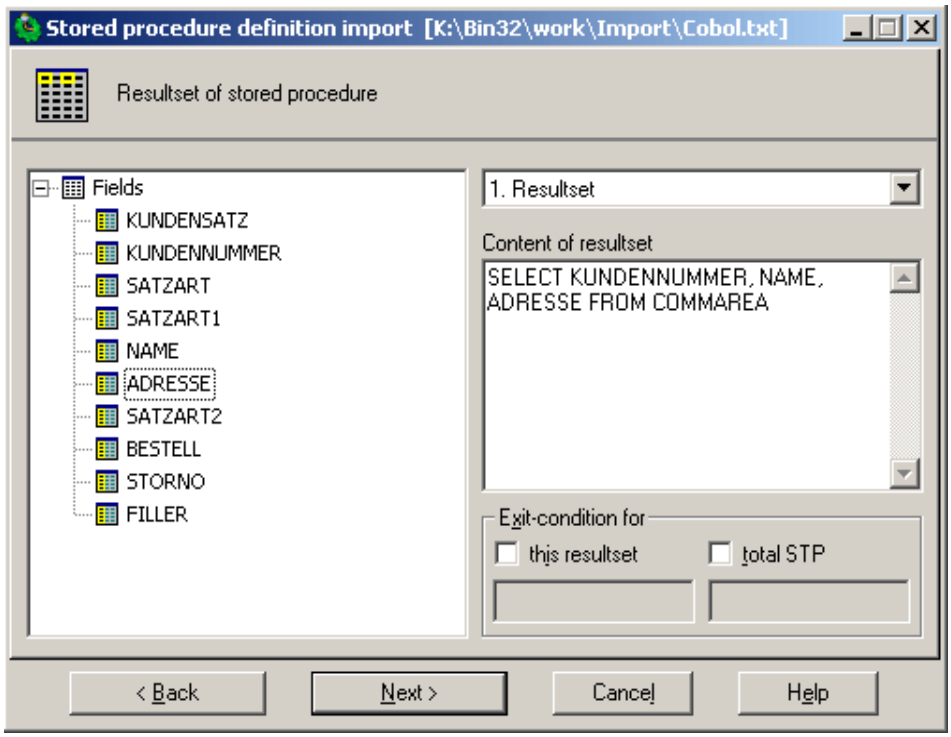


Fig. 34: Import of Stored Procedure definitions – Resultset of Stored Procedure

The dialog components have the following meaning:

Tree view

The tree view displays all available fields of the Stored Procedure. A double click of a field inserts the field into the text field (Content of resultset).

Selection list

Use this lsit field to select the resultset. Entry ”Add a resultset” adds another resultset to the selection list. A maximum of 9 resultsets can be defined.

Text field

Use this text field to define the content of the resultset. In addition to the avauilable fields, expressions can be defined (i.e. `CONCAT('Name: ', NAME)`).

Exit conditions

Activating the optional buttons exit conditions can be defined in the corresponding text fields. The exit condition applies either to the resultset or applies to the full Stored Procedure, meaning that further resultsets will not be executed. As an example: an exit condition can be a certain returncode value returned from the Stored procedure.

2.2.1.2.8 Section "Additional parameter for the Stored Procedure definition on the host"

This dialog only display for the SQL-Engine format. Additional parameter can be defined that may vary for the different data sources.

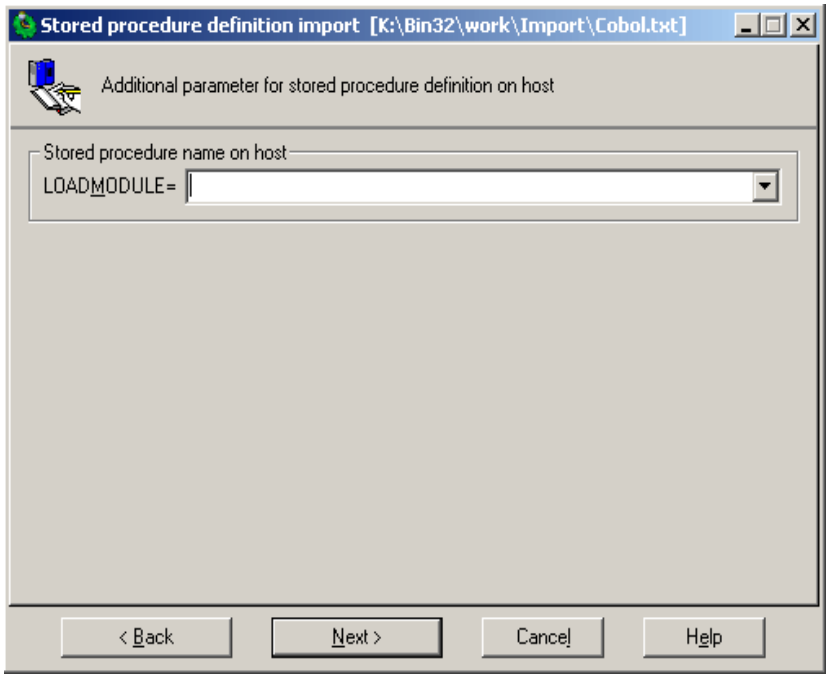


Fig. 35: Import of Stored Procedure definitions - Additional parameter

The dialog components have the following meaning:

Stored procedure name on host

Entry "LOADMODULE=" defines the program that should be executed on the host (Stored Procedure) by the SQL-Engine together with the Stored Procedure definitions.

2.2.1.2.9 Section "Target to receive the imported field definitions"

The result of the import is stored in a file. If a coonnection to the host is active, the file can be saved onto the Virtual Disk. Alternatively the file can also be stored on a PC drive.

Additional information can be found in chapter Section: Target to receive the imported field definitions on page 55 .

2.2.1.3 SQL-Engine Table

The entries on this menu item let you create, modify, or delete the SQL-Engine table definition files. To process SQL accesses by using the SQL-Engine the table description files must be located in .FTB format (or .STP) in a virtual disk directory on the host. The location is determined during the generation (see also the *tcACCESS Customization* manual).

The menu entries available can be used to create, edit, delete and copy the SQL-Engine definition files.

2.2.1.3.1 New definition

This menu item allows you to create a new definition for an SQL-Engine table.

2.2.1.3.2 Edit definition

This dialog allows you to edit SQL-Engine table definitions. You can also load, process, and save a file. It does not matter if the source and the target for this definition file are located on the local PC or the host.

2.2.1.3.2.1 Section: SQL-Engine table definition to be edited

To process an SQL-Engine table definition file, select the file that you want to edit. If a host session is active you may select the file from the virtual disk on the mainframe.

If you are connected with host tcACCESS version 3.01, tcACCESS 3.00 PTF 20 (CICS), or PTF 16 (tcACCESS/VTAM)), the tab dialog Host SQL-Engine definition file displays. A tree view displays, from which you can select a file on the virtual disk for processing.

If the host session is not active, you must define the SQL-Engine definition file on the local PC as the source. The PC file dialog displays when you click the **Next >** button.

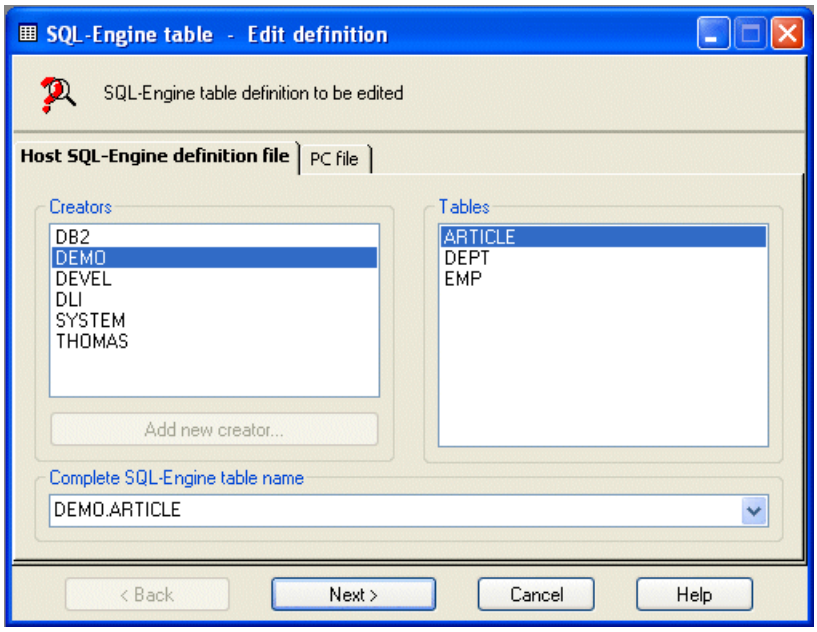


Fig. 36: SQL-Engine table - SQL-Engine table definition to be edited

Host SQL-Engine definition file Tab Dialog:

Tables

All table names associated with the selected creator displays in this window. By selecting an entry on this list an existing view definition is replaced. Each view corresponds to an SQL-Engine table definition file (extension .FTB or .STP) within the creator's sub-directory on the SQL-Engine virtual disk.

Complete SQL-Engine table name

You must enter the complete name of the selected destination table in this field. Only the name entered in this field is used for processing when you click the **Next**> button. The format of an SQL-Engine table name is CREATOR.TABLE.

Host file Tab Dialog:

Tree view

All virtual disks assigned to a local disk drive display in the tree view. The default location for SQL-Engine table definition files (extension: .FTB or .STP) is the directory TABLES of the virtual disk *TCAVDI.SYSDSK*.
When you select a file from the tree view, the name of the file displays in the Host file field. Only the file name in this field is used for processing.

Host file

The SQL-Engine table definition file (extension: .FTB) defined in this field is used to write the field definitions results. The following syntax must be used:

Logic files in virtual disks

	DESCRIPTION
Syntax	A colon separates the name of the virtual disk from the name of the logic file. Directories may be separated by "/" or "\".
Examples	TCAVDI.SYSDSK:/TABLES/JOE/TEST.FTB virtual disk within a VSAM file TCAVDSK VDISK D:/TABLES/JOE/TEST.FTB virtual disk within a CMS file

PC file Tab Dialog

PC file name

The SQL-Engine table definition file (extension: .FTB) defined in this dialog is used for processing. When you click the **Search** button, a dialog displays and allows you to scroll through the directory

2.2.1.3.2.2 Section: Options for SQL-Engine table definition

When the SQL-Engine table definition file is read successfully, you can define settings for additional processing in this section.
You can switch between the current and the next or previous page by clicking the **Next>** or **<Back** buttons. Any changes that you make are immediately accepted when you click the **Next >** button.

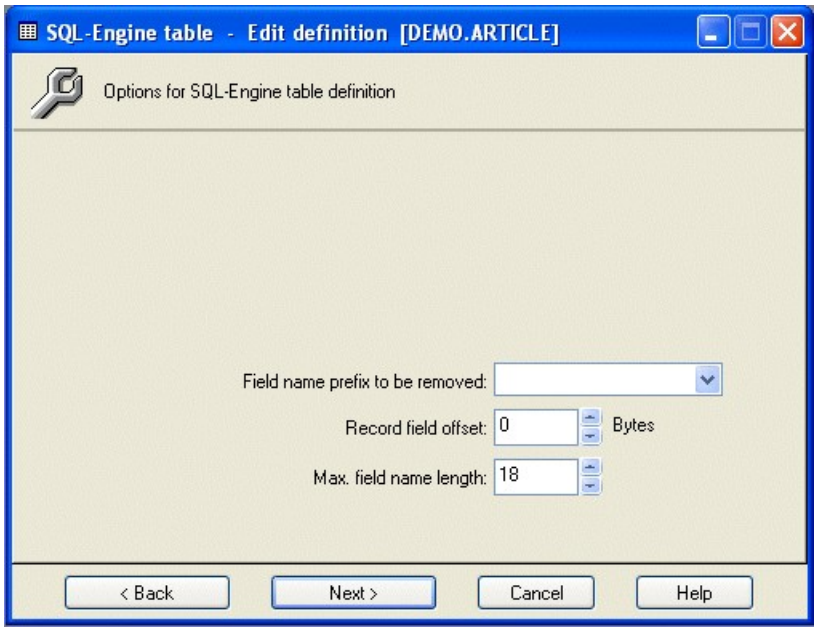


Fig. 37: SQL-Engine table - Options for SQL-Engine table definition

The dialog components are described in the section, , Section: Options for importing field definitions*, page 44.*

2.2.1.3.2.3 Section: Edit field definitions

During this process, all fields in the SQL-Engine table definition file are displayed for processing.
Since hyphens are not allowed within field names in SQL-Engine table definitions, they are replaced by underscores. All field names are truncated to the defined number of positions and made unique.

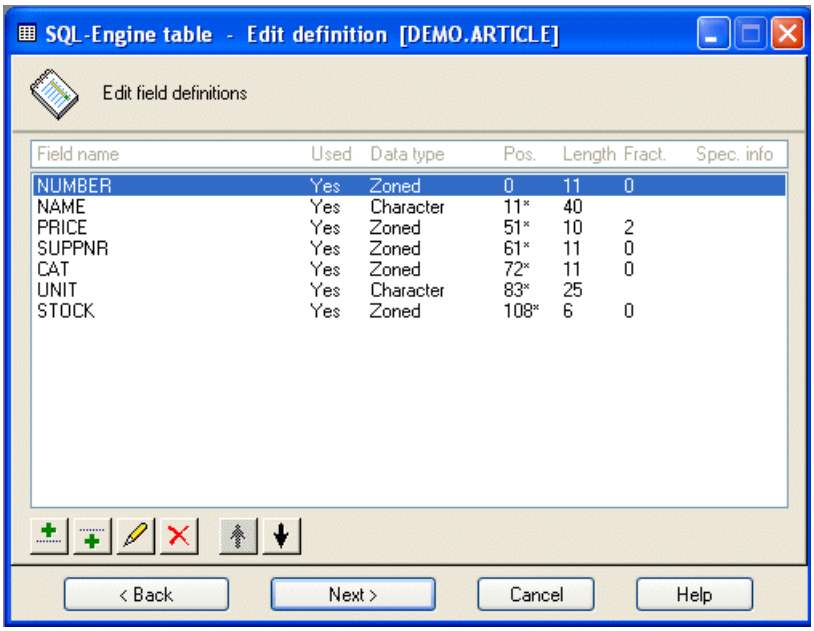


Fig. 38: SQL-Engine table - Edit field definitions

List of field definitions

This list contains all defined fields with their most important criteria. To edit a field, double-click on an entry or click the **Edit** button. Select the **Delete** key to remove the selected list entry. Press the space key to activate or deactivate the USED FOR QUERY option.

The dialog components are described in Section: Check/edit imported field definitions, page 46.

2.2.1.3.2.4 Section: Additional parameters for SQL-Engine table definition

Use this dialog to define additional parameters that may be different depending on the type of data source. The dialog components are described in Section*: Section: Additional parameters for SQL-Engine table definition on host*, page *54*

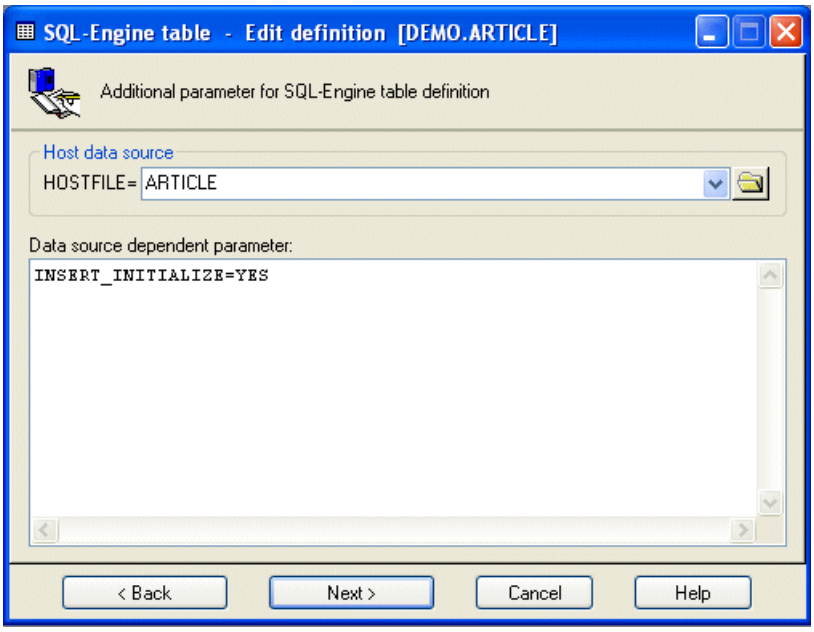


Fig. 39: SQL-Engine table - Additional parameters for SQL-Engine table definition

2.2.1.3.2.5 Section: Write SQL-Engine table definition

Processed results are stored in a target file. This dialog process corresponds to the section Field Definition Import, page *34*.

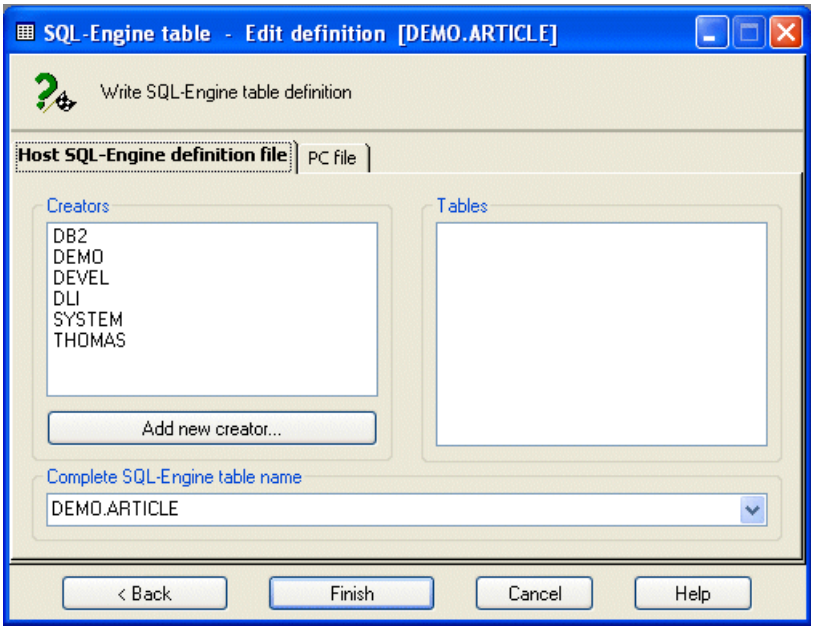


Fig. 40: SQL-Engine table - Write SQL-Engine table definition

When you select the **Finish** button, the target file is written and the dialog closes. The file specification on the tab dialog currently displayed is used.

2.2.1.3.3 Delete definition

This menu item allows you to delete an SQL-Engine table definition file from the host.

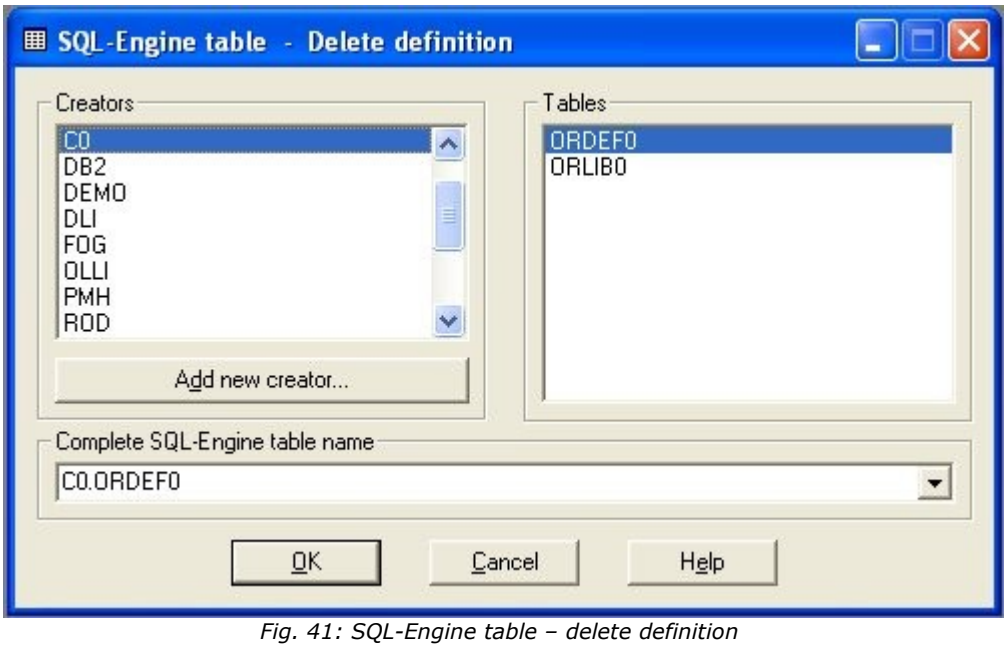


Fig. 41: SQL-Engine table – delete definition

The dialog components have the following meaning:

Creator

The list contains the creator names of all SQL-Engine defined tables. The entries are sub-directories on the SQL-Engine virtual disk. By selecting a creator, all table names belonging to it are retrieved and displayed in the tables window.

Add new Creator

Click this button to insert a new creator. It then provides the required creator name. A new sub-directory on the SQL-Engine virtual disk is created based on the name that has been specified.

Tables

All table names associated with the selected creator displays in this window. By selecting an entry on this list an existing table definition is replaced. Each table belongs to an SQL-Engine table definition file (extension .FTB or .STP) within the creator's sub-directory on the SQL-Engine virtual disk.

Complete SQL-Engine table name

You must enter the complete name of the selected destination table in this field. The Creators and Tables lists provide a reference for the user. The syntax format of an SQL-Engine table name is CREATOR.TABLE.

OK

After a confirmation message the SQL-Engine table definition will be deleted and the dialog terminates.

Cancel

Terminates the dialog an no action is performed

Help

Display the online help for this dialog.

2.2.1.3.4 Copy definition to...

Use this menu item if you want to copy an existing SQL-Engine table definition under a different name. A dialog will be displayed where the old and new CREATOR/NAME can be specified.

The dialog components have the following meaning:

Creator

The list contains all creators of the SQL-Engine. Physically a creator represents a directory on the Virtual Disk for the SQL-Engine. If a creator has been selected, all definitions stored with this creator are displayed in the list to the right.

Add new creator

Click this button to insert a new creator. It then provides the required creator name. A new sub-directory on the SQL-Engine virtual disk is created based on the name that you enter.

Tables

All table names associated with the selected creator display in this window. By selecting an entry on this list an existing table definition is replaced. Each table belongs to an SQL-Engine table definition file (extension .FTB) within the creator's sub-directory on the SQL-Engine virtual disk.

Complete SQL-Engine table name

You must enter the complete name of the selected destination table in this field. The Creators and Tables lists provide a reference for the user. The syntax format of an SQL-Engine table name is CREATOR.TABLE.

Back

Advances to the previous step in the wizard dialog.

Next

Advances to the next step in the wizard dialog.

Cancel

Cancels the dialog. No action is performed.

Help

Displays the online help.

2.2.1.4 SQL-Engine View

SQL-Engine Views are helpful when you get a restricted view on a table or if two tables are combined via a join. You can address the result through the view like a regular table. The view definitions must be in a specific virtual disk directory on the host. This directory is defined during the generation (see also manual tcACCESS Customization). View definitions must have the file extension .VTB.

With the introduction of the "Global Language Pack" in Version 8, all Views are internally stored in UNICODE or UTF-8.

The entries available for the menu support the reation, deletion and update of the SQL-Engine definition files.

2.2.1.4.1 New definition

This menu item allows you to create a new definition for an SQL-Engine view.

2.2.1.4.2 Edit definition

This dialog allows you to edit SQL-Engine table definitions. You can also load, process, and save a file. It does not matter if the source and the target for this definition file are located on the local PC or the host.

2.2.1.4.2.1 Section: SQL-Engine view definition to be edited

To process an SQL-Engine definition file, select the file that you want to edit. If a host session is active you may select the file from the virtual disk on the mainframe.

If you are connected with host tcACCESS version 3.01, tcACCESS 3.00 PTF 20 (CICS), or PTF 16 (tcACCESS/VTAM)), the tab dialog Host SQL-Engine definition file displays. A tree view displays, from which you can select a file on the virtual disk for processing.

If the host session is not active, you must define the SQL-Engine definition file on the local PC as the source. The PC file dialog displays when you click the **Next >** button.

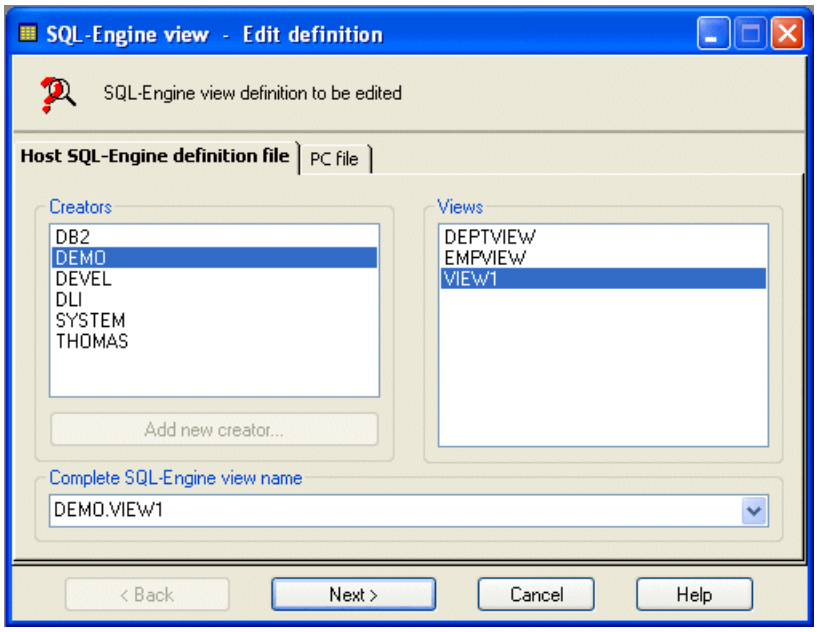


Fig. 42: SQL-Engine view - SQL-Engine view definition to be edited

Host SQL-Engine definition file Tab Dialog:

Views

All view names associated with the selected creator displays in this window. By selecting an entry on this list an existing view definition is replaced. Each view corresponds to an SQL-Engine view definition file (extension .VTB) within the creator's sub-directory on the SQL-Engine virtual disk.

Complete SQL-Engine view name

You must enter the complete name of the selected destination table in this field. Only the name entered in this field is used for processing when you click the **Next**> button. The format of an SQL-Engine view name is CREATOR.VIEW.

Host file Tab Dialog:

Tree view

All virtual disks assigned to a local disk drive display in the tree view. The default location for SQL-Engine view definition files (extension: .VTB) is the directory TABLES of the virtual disk *TCAVDI.SYSDSK*.
When you select a file from the tree view, the name of the file displays in the Host file field. Only the file name in this field is used for processing.

Host file

The SQL-Engine view definition file (extension: .VTB) defined in this field is used to write the field definitions results. The following syntax must be used:

Logic files in virtual disks

	DESCRIPTION	
Syntax	A colon separates the name of the virtual disk from the name of the logic file. Directories may be separated by "/" or "\".	
Examples	TCAVDI.SYSDSK:/TABLES/JOE/TEST.VTB	virtual disk within a VSAM file
	TCAVDSK VDISK D:/TABLES/JOE/TEST.VTB	virtual disk within a CMS file

PC file Tab Dialog

PC file name

The SQL-Engine view definition file (extension: .VTB) defined in this dialog is used for processing. When you click the **Search** button, a dialog displays and allows you to scroll through the directory

2.2.1.4.2.2 Section: Edit SQL command to be executed

You can edit the SQL command that is executed when the view is applied..

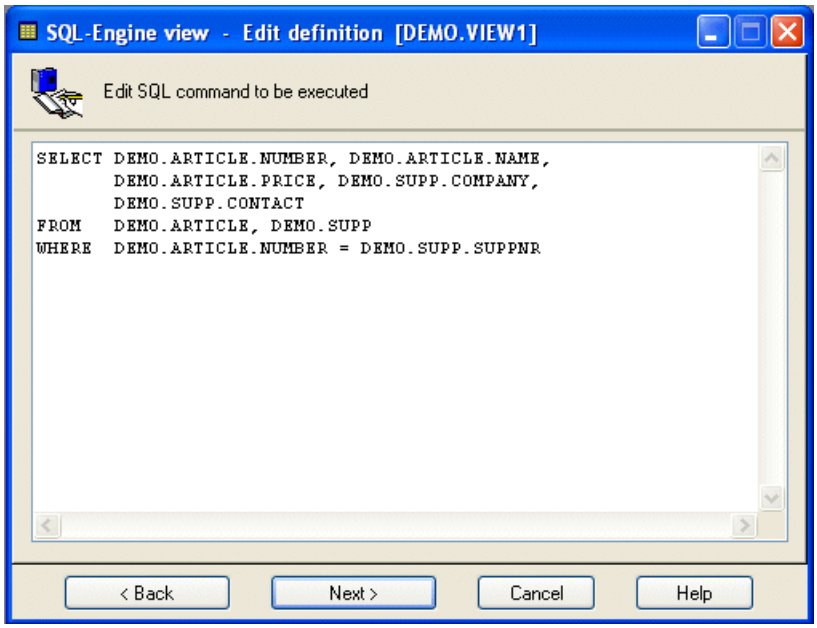


Fig. 43: SQL-Engine view - Edit SQL command to be executed

The dialog components are described below:

SQL command

The SQL command that displays in the dialog window is ecuted when view is applied via the SQL-Engine (e.g., select * from CREATOR.VIEW). It is a select command that determines the data. You can use drag and drop to enter the content of a text file in this field.

2.2.1.4.2.3 Section: Write SQL-Engine view definition

Processed results are stored in a target file. This dialog process corresponds to the Section: SQL-Engine view definition to be edited, page 72.

When you click the **Finish** button, the system writes the target file and the dialog closes. The file specification on the tab dialog currently displayed is used.

2.2.1.4.3 Delete definition

This menu item allows you to delete an SQL-Engine view definition file from the host.

2.2.1.4.4 Copy definition

This menu entry can be used to copy an existing veiw definition file under a new name.A dialog will be displayed that asks for the old and new creator/name.

The dialog components have the following meaning:

Creator

The list contains all creators of the SQL-Engine. Physically a creator represents a directory on the Virtual Disk for the SQL-Engine. If a creator

has been selected, all definitions stored with this creator are displayed in the list to the right.

Add new creator

Click this button to insert a new creator. It then provides the required creator name. A new sub-directory on the SQL-Engine virtual disk is created based on the name that you enter.

Views

All view names associated with the selected creator displays in this window. By selecting an entry on this list an existing view definition will be copied. Each view corresponds to an SQL-Engine view definition file (extension .VTB) within the creator's sub-directory on the SQL-Engine virtual disk.

Complete SQL-Engine view name

You must enter the complete new name of the selected destination view in this field. Only the name entered in this field is used for processing when you click the **Next**> button. The format of an SQL-Engine view name is CREATOR.VIEW.

< Back

Advances to the previous dialog step of the wizard.

Next >

Advances to the next dialog step of the wizard

Cancel

Terminates the dialog. No changes will be saved.

Help

Displays the online help for the dialog.

2.2.1.5 SQL-Engine Stored Procedure

You can apply stored procedures to start a host program by using an SQL command. Use stored procedures to retrieve data, perform table calculations, etc., and then return the result to the calling system. Stored procedures must be located within a specific directory on the host virtual disk, which is determined during the generation (see also manual tcACCESS Customization). They must have an .STP file extension.

2.2.1.5.1 New definition

This menu item allows you to create a new definition for an SQL-Engine stored procedure.

2.2.1.5.2 Edit definition

This dialog allows you to edit SQL-Engine stored procedure definitions. You can also load, process, and save a file. It does not matter if the source and the target for this definition file are located on the local PC or the host.

2.2.1.5.2.1 Section: SQL-Engine stored procedure to be edited

To process an SQL-Engine definition file, select the file that you want to edit. If a host session is active you may select the file from the virtual disk on the mainframe.

If you are connected with host tcACCESS version 3.01, tcACCESS 3.00 PTF 20 (CICS), or PTF 16 (tcACCESS/VTAM)), the tab dialog Host SQL-Engine definition file displays. A tree view displays, from which you can select a file on the virtual disk for processing.

If the host session is not active, you must define the SQL-Engine definition file on the local PC as the source. The PC file dialog displays when you click the **Next >** button.

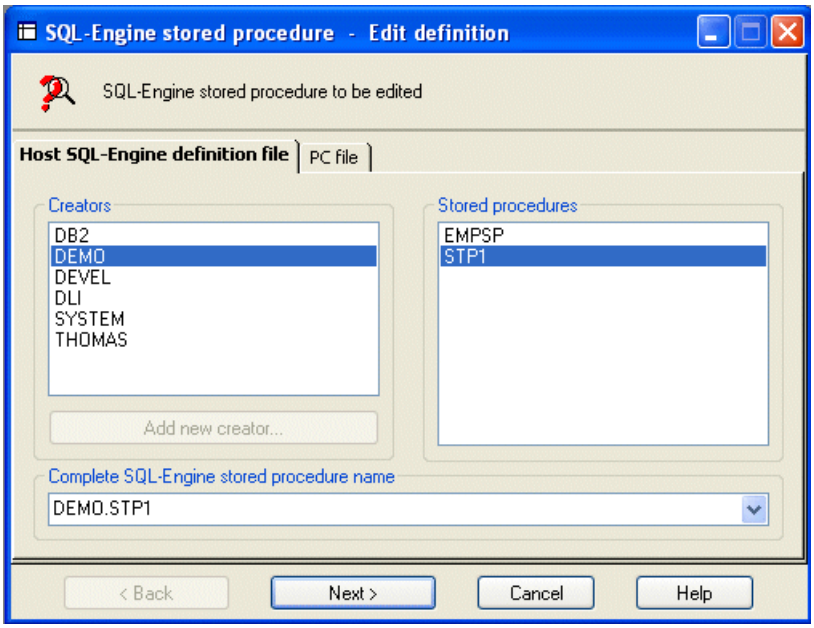


Fig. 44: SQL-Engine Stored Procedure - SQL-Engine Stored Procedure to be edited

Host SQL-Engine definition file Tab Dialog

Stored Procedures

All stored procedures associated with the selected creator are displayed in this list. To edit an existing stored procedure file, select the file. Each table corresponds to an SQL-Engine stored procedure file (extension: .STP) within the creator sub-directory on the SQL-Engine virtual disk.

Complete SQL-Engine stored procedure name

Enter the name of the stored procedure in the edit field. Click the **Next**> button. Only the entry in this field is scanned for processing. The format of an SQL-Engine stored procedure is CREATOR.PROCEDURE.

Host file Tab Dialog

Tree view

All virtual disks assigned to a local disk drive display in the tree view. The default location for SQL-Engine table definition files (extension: .STP) is the directory TABLES of the virtual disk *TCAVDI.SYSDSK*.

Host file

The SQL-Engine table definition file (extension: .STP) defined in this field is used to write the resulting field definitions. The following syntax must be used:

Logic files in virtual disks

	DESCRIPTION	
Syntax	A colon separates the name of the virtual disk from the name of the logic file. Directories may be separated by "/" or "\".	
Examples	TCAVDI.SYSDSK:/TABLES/JOE/TEST.STP	virtual disk within a VSAM file
	TCAVDSK VDISK D:/TABLES/JOE/TEST.STP	virtual disk within a CMS file

PC file Tab Dialog

PC file name

The SQL-Engine stored procedure definition file (extension: .STP) defined in this field is used as input for additional processing. When you click the **Search** button, a dialog displays and allows you to scroll through the directory.

2.2.1.5.2.2 Section: Edit SQL-Engine stored procedure

This procedure allows you to edit the stored procedure.

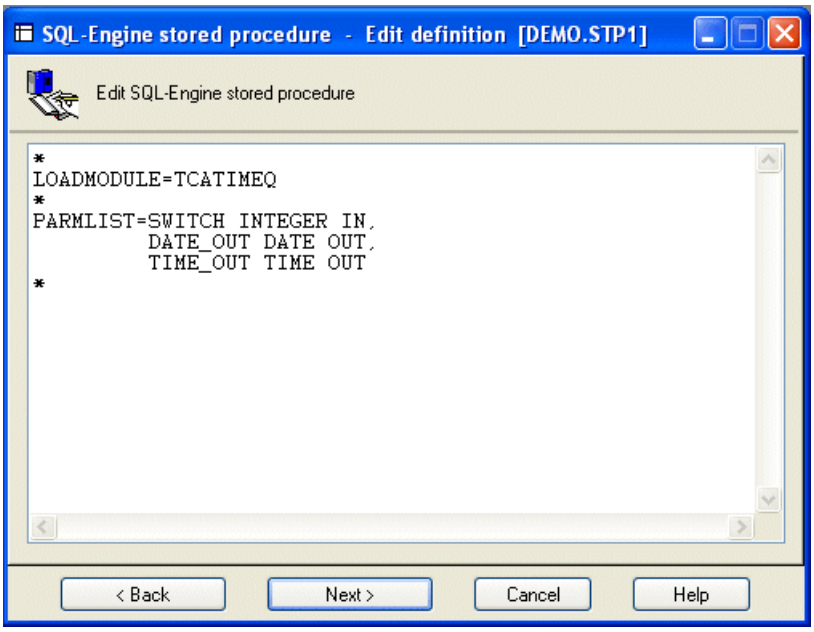


Fig. 45: SQL-Engine stored procedure - Edit SQL-Engine stored procedure

The dialog components are described below:

Stored procedure

Use this field to enter the name LOADMODULE= and the input/output parameter PARMLIST= of the stored procedure, which is processed. You can use drag and drop to move the content of a text file into this field.

If a host session exists to a Version 8 tcACCESS host, the Stored Procedure is processed using the new format. All related steps are are described in section Edit field definition on page 62. A Stored Procedure using the old format is converted into the new format.

2.2.1.5.2.3 Section: Write SQL-Engine Stored Procedure

The result of the conversion is saved in the target file. Instruction for this dialog processing is described in Section: SQL-Engine stored procedure to be edited (page 76).

When you click the **Finish** button, the system writes the target file and closes the dialog. The file specification on the active tab dialog is used for processing.

2.2.1.5.3 Delete definition

This menu item allows you to delete an SQL-Engine stored procedure definition file from the host.

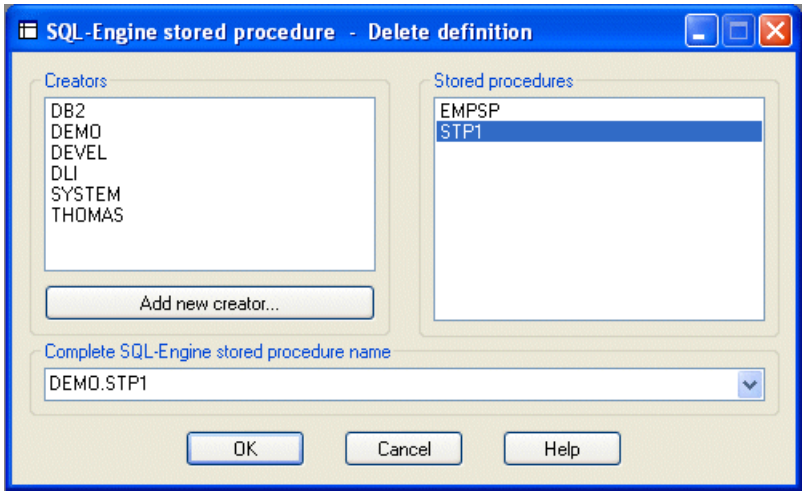


Fig. 46: SQL-Engine Stored Procedure – Delete definition

The dialog components have the following meaning:

Creator

The list shows all creators in the SQL-Engine. A creator is the name of a directory on the Virtual Disk for the SQL-Engine. If a creator is being selected and Stored Procedures definitions exist in this directory, they will be displayed.

Add new creator

Click this button to add a new creator. Provide the required creator name. A new sub-directory on the SQL-Engine virtual disk is created based on the name that you enter.

Stored Procedures

All Stored Procedure definitions for the selected creator will be displayed. If a definition should be deleted, simply select it from the list. A Stored Procedure definition is a file on the Virtual Disk with an extension of ".STP".

Full and qualified name of the SQL-Engine Stored Procedure

Specify the fully qualified name of the Stored Procedure definition that is to be deleted. The name of the SQL-Engine Stored Procedure has the format "CREATOR.STOREDPROC".

OK

You will be asked to confirm the deletion of the specified SQL-Engine Stored Procedure definition, the definition will be deleted and the dialog box will be closed.

Cancel

Terminates the dialog, no action will be taken.

Help

Displays the online help for the dialog.

2.2.1.5.4 Opy definition

Ithis menu allows to copy a Stored Procedure definition using a different name. A dialog is displayed that sks for the old and new Creator/name. The definition is copied.

The dialog components have the following meaning:

Creator

The list contains the creator names of all SQL-Engine defined tables. The entries are sub-directories on the SQL-Engine virtual disk. By selecting a creator, all table names belonging to it are retrieved and displayed in the tables window.

Add new creator

Click this button to insert a new creator. It then provides the required creator name. A new sub-directory on the SQL-Engine virtual disk is created based on the name that you enter.

Stored Procedures

All Stored Procedures for the creator are displayed. Toc opy an entry it must be selected from the list. Each entry represents a SQL-Engine Stored procedure definition file (extension .STP). It is stored in the subdirectory of the corresponding creator on the Virtual Disk.

Complete name of the SQL-Engine Stored Procedure

The full name must be specified of the SQL-Engine Stored Procedure that is to be copied. Only this filed is validated when the OK butoon is pressed. The list of ”creators” and ”Stored Procedures“ are only used to simplify the input. A Stored Procedure name has the following format: ”CREATOR.STOREDPROC“.

< Back

Displays the preceding import step. Settings and/or selections may be changed.

Next >

Proceed to the next dialog step (section).

Cancel

Terminates the dialog, no action is performed.

Help

Displays the online help for the dialog.

2.2.1.6 SQL-Engine Precompiled Commands

Using the Precompiled Commands feature allows a timely separation of compilation and execution of an SQL-statement. The compilation and the preparation of the structure- and index-definitions of the SQL-statement is only performed once. The compiled object can then be used and executed any number of times without having to go through the compile effort again. The overall processing time will be reduced, especially for those statements, where the time spent in compiling the statement is high compared to the execution time.

The Precompiled Commands will be stored in a reserved part of the Virtual Disk on the host. This part has been defined during the system generation of tcACCESS (refer to manual "tcACCESS Customization"). The file for the Precompiled Commands have an extension of ".CTB".

This menu item and its submenus can be used to create, change and delete the SQL-Engine Precompiled Command-files

2.2.1.6.1 New definition

Use this menu item to create an new Precompiled Command-file. The usage of this menu item is identical to the EDIT dialog as described in Section "Edit of an SQL-Engine Precompiled Command" (page 86).

2.2.1.6.2 Edit definition

Use this menu item to load, edit and save an existing definition of a Precompiled Command. The source and the target for the definition file can reside on a PC or on the host.

The buttons at the bottom of the dialog have the following meaning:

< Back

Displays the preceding step. Settings and/or selections may be changed.

Next > / Finish

As long as the last dialog window has not yet reached, the button reads "Next >", otherwise it reads "Finish". Use this button to step through the dialog windows or to complete the definitions.

Cancel

Terminate the dialog with no action taken.

Help

Display the online help for this dialog.

2.2.1.6.2.1 Section "Prepare the edit of an SQL-Engine Precompiled Command"

To edit a definition for an SQL-Engine Precompiled Command you must select the desired file. If a connection to the host exists the file may also exist on the Virtual Disk of the mainframe.

The tab "Host SQL-Engine definition file" will be displayed where you can specify the name of the file (in format "CREATOR.PRECOMPILEDCOMMAND").

If no connecrtion to the host exists, the SQL-Engine definition file must exist on the local PC. In general: when clicking the "Next ->"-button, the tab will be used in the dialog that currently occupies the foreground view.

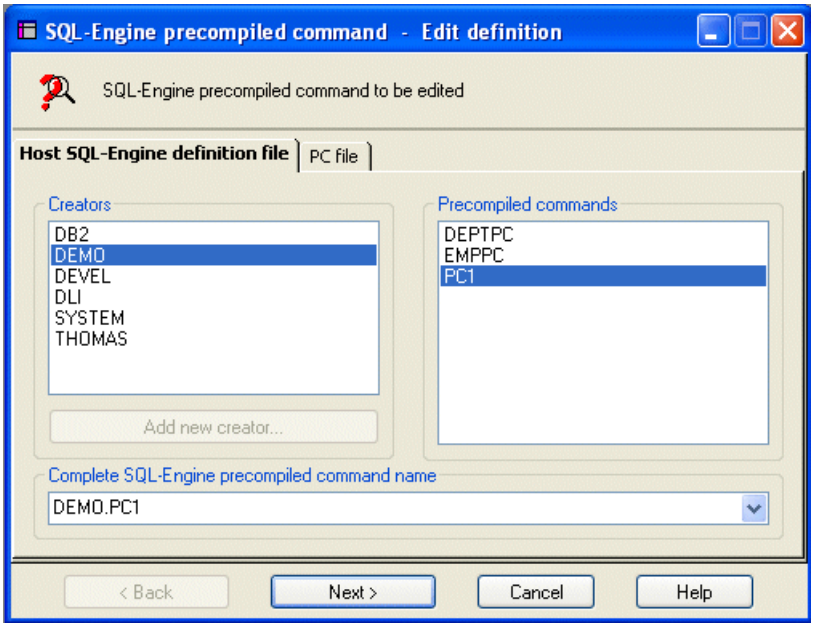


Fig. 47: SQL-Engine Precompiled Command – Edit dialog

The dialog components have the following meaning:

Tab "Host SQL-Engine definition-file"

Creators

The list shows all creators in the SQL-Engine. A creator is the name of a directory on the Virtual Disk for the SQL-Engine. If a creator is being selected and Precompiled Command definitions exist in this directory, they will be displayed.

Precompiled Commands

All Precompiled Command-definitions for the selected creator will be displayed. If a definition should be edited, simply select it from the list. A Precompiled Command definition is a file on the Virtual Disk with an extension of ".CTB".

Full and qualified name of the SQL-Engine Precompiled Command

Specify the fully qualified name of the Precompiled Command definition that is to be deleted. The name of the SQL-Engine Precompiled Command has the format "CREATOR.PRECOMPILEDCOMMAND".

Tab "PC-file"

PC-file name

The specified file name for the SQL-Engine Precompiled Command-file (Extension: ".CTB") will be used as source and/or target. Press the BROWSE button to browse for the file.

Note: A SQL-Engine Precompiled Command-file that is saved on the PC differs from the corresponding host file. The PC file contains the SQL-statement in a readable format, where the host-file already contains the compiled format.

2.2.1.6.2.2 Section "Edit of an SQL-Engine Precompiled Command"

In this dialog the SQL statement can be edited. Detailed information about the SQL-syntax can be found in the manual "tcACCESS Host Server".

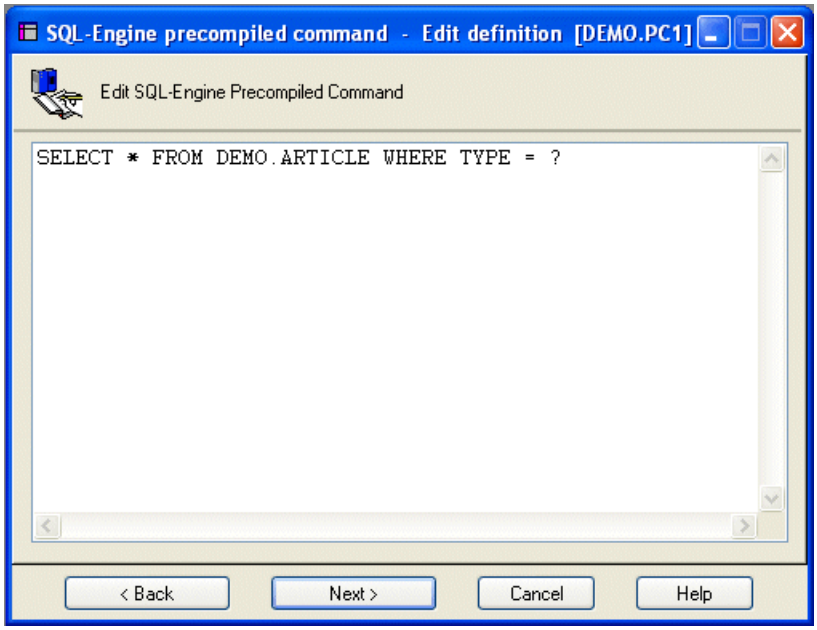


Fig. 48: SQL-Engine Precompiled Commands- Command edit

The dialog components have the following meaning:

Precompiled Command

Use this field to specify the SQL-statement that should be saved as Precompiled Command. You can use "Drag and Drop" to drop a text-file into the field which is then opened and can be edited.

2.2.1.6.2.3 Section "Write the SQL-Engine Precompiled Command"

The result of the edit will be saved into the target file. This dialog-part is identical to Section "Prepare the edit of an SQL-Engine Precompiled Command" (page 84).

"Finish" will write the target-file and the dialog will be closed. The file name of the tab that occupies the foreground view will be used.

2.2.1.6.3 Delete definition

Use this menu item if a Precompiled Command-definition of the SQL-Engine should be deleted.

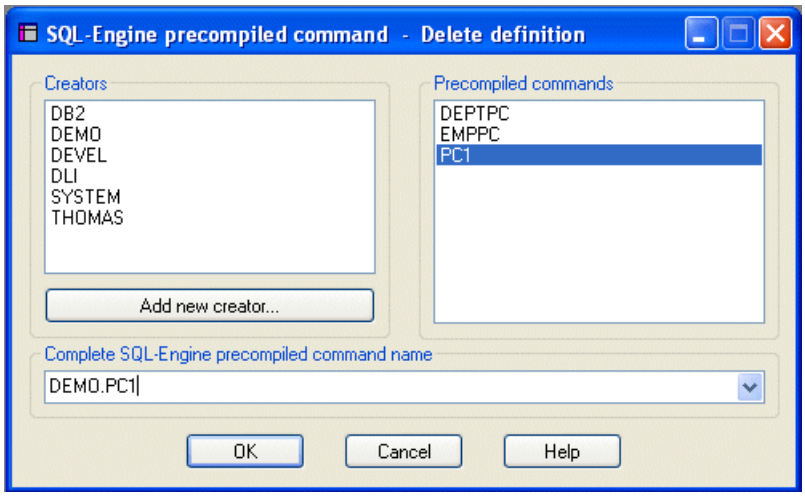


Fig. 49: SQL-Engine Precompiled Command – Delete definition

The dialog components have the following meaning:

Creator

The list shows all creators in the SQL-Engine. A creator is the name of a directory on the Virtual Disk for the SQL-Engine. If a creator is being selected and Precompiled Command definitions exist in this directory, they will be displayed.

Add new creator

Click this button to insert a new creator. Provide the required creator name. A new sub-directory on the SQL-Engine virtual disk is created based on the name that you enter.

Precompiled Commands

All Precompiled Command-definitions for the selected creator will be displayed. If a definition should be edited, simply select it from the list. A Precompiled Command definition is a file on the Virtual Disk with an extension of ".CTB".

Full and qualified name of the SQL-Engine Precompiled Command

Specify the fully qualified name of the Precompiled Command definition that is to be deleted. The name of the SQL-Engine Precompiled Command has the format "CREATOR.PRECOMPILEDCOMMAND".

OK

You will be asked to confirm the deletion of the specified SQL-Engine Precompiled Commnad definition, the definition will be deleted and the dialog box will be closed.

Cancel

Terminates the dialog, no action will be taken.

Help

Displays the online help for the dialog.

2.2.1.6.4 Copy definition

This menu can be used to copy an existing Precompiled Command-definition file using a new name. A dialog is displayed that asks for the old and new name.

The dialog components have the following meaning:

Creator

The list contains the creator names of all SQL-Engine defined tables. The entries are sub-directories on the SQL-Engine virtual disk. By selecting a creator, all table names belonging to it are retrieved and displayed in the Tables window.

Add new creator

The list contains the creator names of all SQL-Engine defined tables. The entries are sub-directories on the SQL-Engine virtual disk. By selecting a creator, all table names belonging to it are retrieved and displayed in the Tables window.

Precompiled Commands

All Precompiled Commands of the selected creator are displayed. To copy a command select the source entry from the list. Each entry represents a SQL-Engine Precompiled Command file (Extension: „.CTB"). It is stored in the subdirectory of the corresponding creator on the Virtual Disk.

Complete SQL-Engine precompiled command name

Specify the full name of new precompiled command file. This name will be used when the "OK"-button is pressed. A SQL-Engine Precompiled Command name has the following format "CREATOR.PRECOMPILEDCOMMAND".

< Back

Proceeds one step back in the dialog sequence.

Next >

Proceeds one step forward in the dialog sequence.

Cancel

Termninates the dialog, no action is performed.

Help

Display the online help for the dialog.

2.2.1.7 Change language

When you select this menu item, a list of the installed language modules displays.

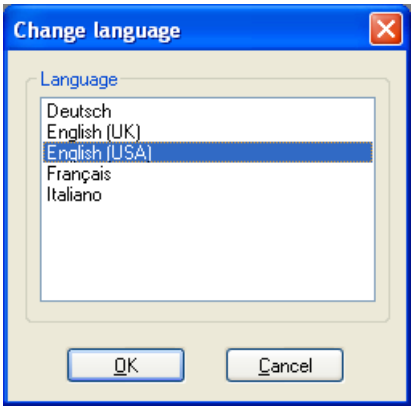


Fig. 50: Change language dialog

The language currently in use is selected when the dialog is called. When you select another language and confirm by clicking the **OK** button, the language for all tcACCESS displays changes.

2.2.1.8 Exit

Selecting **Exit** terminates the tcACCESS program. You can also terminate tcACCESS by clicking the following icon.

If a host session is active when you select Exit, the system asks whether the host session also ends. Answer no if additional host access is required (e.g., via a batch program).

2.2.2 The Session Menu

The Session menu provides functions for starting and ending a session and entering program settings.

2.2.2.1 Allocate Session

To allocate a session, select the appropriate menu item or click the icon shown below. A session will be established using the actual connection slot. The

currently active slot is displayed at the end of the list of all defined slots in the "Session" Menu. If a session established successfully, the icon appears pressed. Clicking the icon again ends an opened session.

For an overview on host connection types see Connection configurations, page 190 and Supported Types of Host Connections, page *1*).
Switching the currently used connection slot changes the status of the icon., this always represents the status of the actual slot. If a session has been established successfully, a message will be displayed in the status line, similar to "Slot n (*configuration)* Session allocated to CICS *"Applid" (OS/390|VSE)*.

The buttons on the dialog are described below.

OK

When you click this button, the system attempts to start a session on the host based on the defined host connection (Connection configurations, page 190).

Cancel

Selecting this button ends the dialog without performing any operation.

Help

Displays online help for the opened dialog.

Tab Dialog User logon

Use this tab dialog to define a user name and a password for a mainframe logon. An appropriate mainframe configuration is necessary.

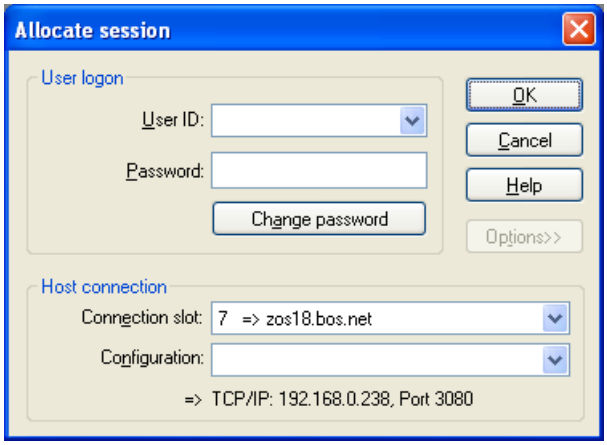


Fig. 51: Allocate Session - Tab Dialog "User Logon"

The dialog components are described below.

User ID

Use this field to enter a user name, which must be defined on the host. Also, you can select a name from the combo box, which lists the most recently used names.

Password

Use this field to enter the password for the user ID above. Each field entry displays as an asterisk ("*").

Options>>

This button can only be used, if the option "Show connection info" has been selected (refer to Host Connections on page 192). The sign on dialog will be expanded by group ""Host connection" (See below).

Host Connection

If option "Options>>" has been selected additional elements will be displayed in the dialog. The defined connection slots as well as the vailable configurations will be displayed.

If option "Allow connection change" has been set, the user can actually select the slot or the configuration to be used.. I fa slot has been selected, tcACCESS tries to establish a connection using this slot and ist assigned configuration. If, however, a configuration has been selected, tcACCESS tries to use the next available slot assigned to this configuration to establish the connection.

2.2.2.1.1 Change password

If the entries on the User logon dialog are valid, you can change the user's host password.

Fig. 52: Allocate Session - "Change Password"

The dialog components are described below.

New password

Use this field to enter the new password for a host logon. The User Logon dialog must contain valid entries. You must also enter the new password in the "Verify new password" field.

Verify new password

As long as the character string input of this field is not identical with the input of the "New password" field, the **OK** button is not available. If the entry is correct, the **OK** button is available.

2.2.2.2 Deallocate Session

End an open session by selecting the appropriate menu item, or click the toolbar icon shown below. This stops the current session and the connection to the host is deallocated.

The PC Communication Handler remains loaded in the background and waits for the establishment of a new connection. When a session ends, the user must close the PC Communication Handler.

2.2.2.3 Options

Open the Options dialog by selecting the appropriate menu item or click the button shown below. This dialog allows the user to change various program settings. An administrator logon is not necessary.

2.2.2.3.1 Tab "Display"

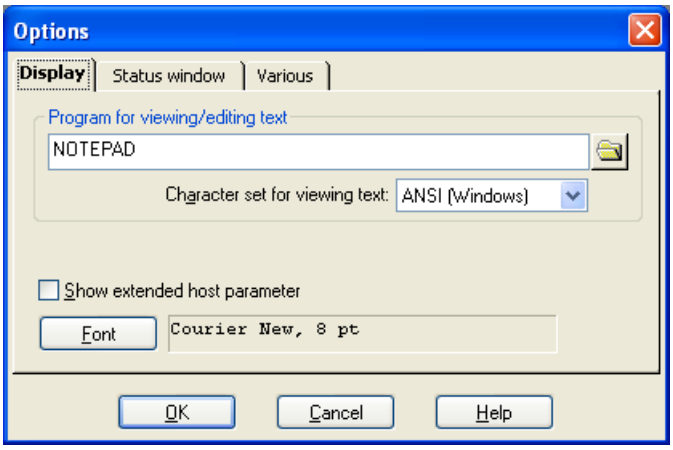


Fig. 53: Options Tab Display

The dialog components are described below.

Program for viewing text

If no PC file is defined as target file for a Host => PC Transfer,(see page *127*)), for Host SQL Query (see page 156), for processing a VM/CP command via the VM/CP Command Tab Dialog , page *121*, the transfer data is saved in a temporary file within the WORK directory as $Txxxxxx.DAT. When the data transfer is successful, the program specified in the Program for viewing text field starts with the temporary file as the parameter. These temporary files are deleted when tcACCESS ends.

Character set for viewing text

Use the selected character set display only in combination with the specified PROGRAM FOR VIEWING TEXT. For example: use a DOS program to display host data, and the MS Windows character set to display normal data transfer.

Zoom-Factor

The "Zoom factor" is used to change the size of dialog bos and their elements. If your monitor supports a high resolution, this factor can be set to increase the size of the dialog box.

Show extended host parameters

Some rarely-used parameters of the PC => Host Transfer *(page 97),* Host => PC Transfer (see page *127)* and for Host SQL Query (see page *156*) dialogs reside on their Ext. Host parameters tab dialogs. For ease of use, the tab dialogs appear only when this option is active.

Font

Select a different font and attributes for the input window of the Host-SQL query (refer to Host SQL Query page 156).

2.2.2.3.2 Tab "Status Window"

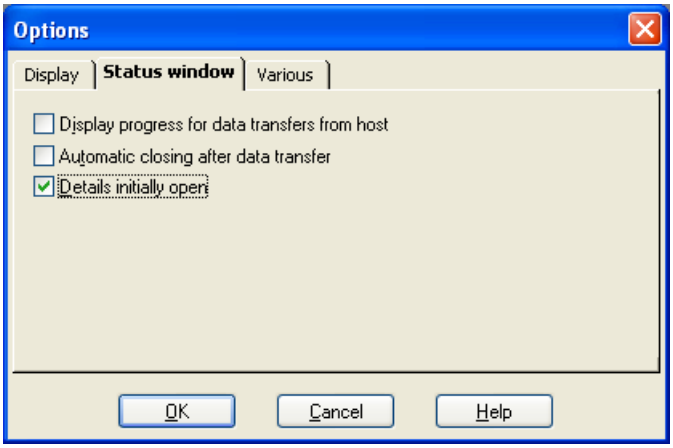


Fig. 54: Options Tab Status Window

Progress display for data transfer from host

When this option is selected, a proportional progress display is used for file transfer from the mainframe.

When this option is active, it may be necessary to perform the data retrieval on the host twice (the first time to count the resulting data, the second time to get the data).

The following data sources are read twice:

- VSAM KSDS, ESDS, RRDS
- TS Queues in combination with SQL selection requests
- MVS PDS/PS
- VSE Library
- VSE Library (Catalog)

Automatic closing (of status window) after data transfer

For the functions PC => Host Transfer *(page 97),* Host => PC Transfer (see page *127)* and for Host SQL Query (see page *156*), a status window is displays during the data transfer. If this option is active, the status window closes automatically at the end of the transfer.

Details (of status window) initially open

If this option is active during a data transfer, all status windows open with detailed display.

2.2.2.3.3 Tab "Various"

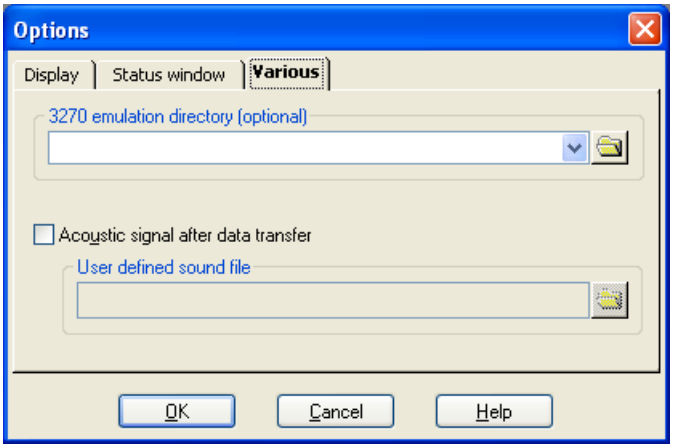


Fig. 55: Options Tab Status Window

3270 emulation directory (optional)

Only the HLLAPI library of the applied 3270 emulation is located, if there is an entry in the search path of the system (environment variable PATH). To avoid this, or if there is a risk that the incorrect library may be loaded due to the existence of several 3270 emulations, use this field to enter a directory, which searches before the system search path**.**

Acoustic signal after data transfer

If this option is active, a tone or a sound file plays to indicate that the transfer of a file between host and PC is complete. Also, the reception of an SQL query result is signaled. An acoustic signal indicates that a data transfer has failed or is incomplete (e.g., because of a connection error).

User-defined sound file

If this field is empty, the PC speaker signals the end of a data transfer. If the field contains the path to a sound file (extension: .wav), this plays instead (requires sound card).

OK

When you click the **OK** button the modified settings are saved and the dialog closes.

2.2.2.4 Explorer Extension Setup

The tcACCESS extension of the Windows Explorer allows you to access a virtual disk on the host. You can customize this feature using the following dialog:

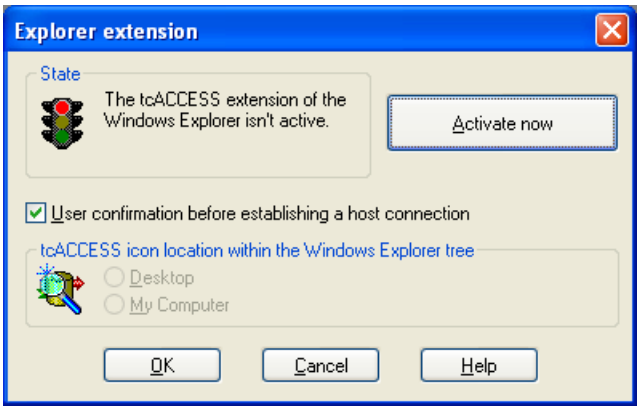


Fig. 56: Explorer extension setup

The dialog components are described below.

State

The current operation mode of the tcACCESS Explorer extension displays in this section.

Activate now / Deactivate now

This button allows you to activate or deactivate the Explorer Extension. The user must have administrator authorization to complete this operation in Windows NT/2000. Under Windows Vista the main program must already be started with administrator permissions.

User confirmation before establishing a host connection

If this option is selected, a confirmation is requested before a connection to the host is established in order to avoid any undesired network activities.

tcACCESS icon location within the Windows Explorer tree

Use this symbol to access virtual disks. Locate this symbol on the Desktop or in the My Computer group. Under Windows NT/2000, the user must have administrator authorization to switch the icon's location.

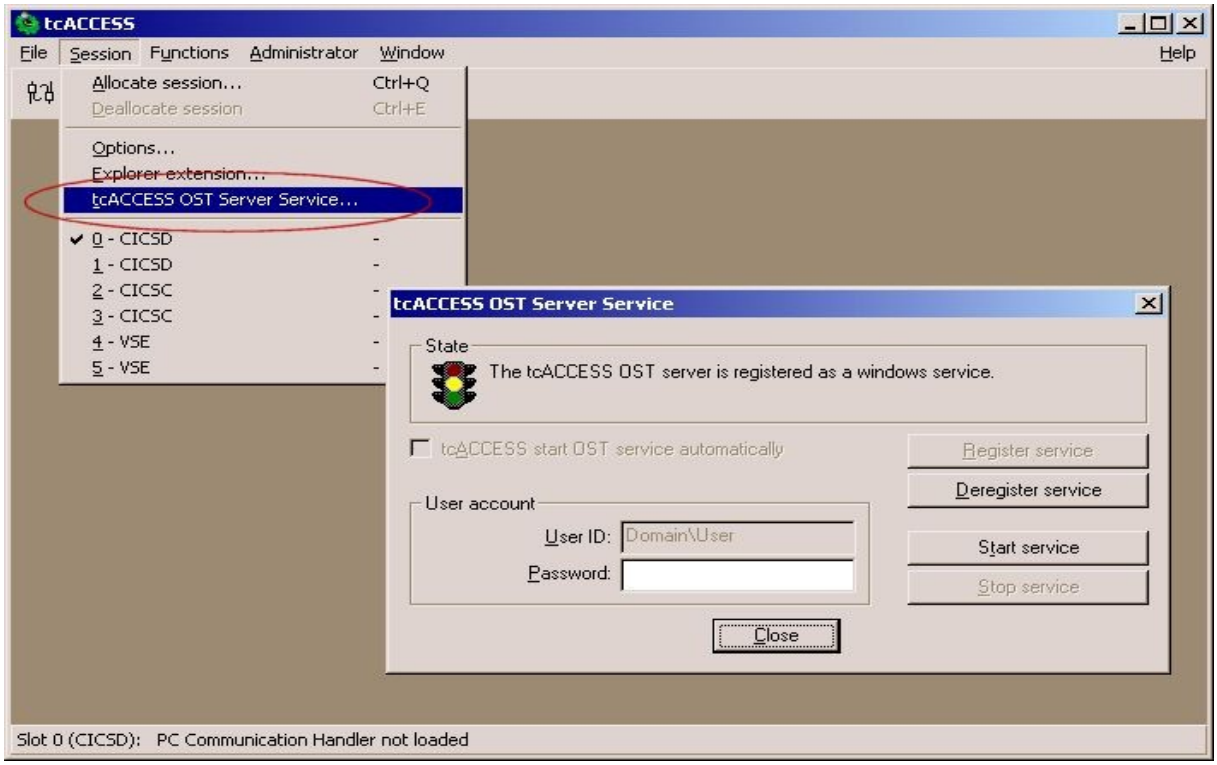
OK

When you click this button the modified settings are saved and the dialog closes.

2.2.2.5 The OST Server Service

The current distribution of tcACCESS contains a server component that supports access to Open System databases. All types of DB/2 UDB, all ODBC data sources and Oracle (as of version 9i) are supported. Achieving best performance has been implemented into the system.

The OST Server can be installed and started as a service using this menu entry. The advantage of the usage as a service is because there are no interactive communications with the server to use this tcACCESS component.



The tcACCESS OST Server uses port 3020 for the listener. If another port should be used, the following entry in file "tcACCESS.ini" of the user must be changed:

```
[TCASOSTQ]
Port=nnnnn
```

A description of the usage of OST and required parameter can be found in manual "tcACCESS Host Server".

2.2.2.6 "Slot" Selection

All slots assigned to a connection configuration are displayed at the end of the "Session" menu. If the slot is being marked this indicates the active slot. This slot can be changed at any time. An active host session remains active.

2.2.3 Functions Menu

The Functions menu includes all tcACCESS facilities for exchanging data between PC and host.

2.2.3.1 PC => Host Transfer

Open the PC => Host Transfer dialog by selecting the appropriate menu item or by clicking the button as shown below This function allows you to transfer data from a local PC to a mainframe (upload). tcACCESS supports a number of host file types, e.g., VSAM, MVS PDS/PS, TS Queues and Print Queues.

A host session must be started before this menu item can be selected (see Allocate Session, page *89*).
The input fields are initialized internally by the program. However, the start values for the input fields may also be defined individually by pressing the Load/Save Parameters button and selecting the menu entry Set as default.

The buttons on the dialog are described below:

View / Execute

Click this button to start the data transfer from a PC to the host. The system checks the validity of the parameters in advance. When an error is detected a message displays. If all values are correct, a status window displays at the beginning of the transfer (see, page *269*).

When the data transfer is complete the dialog remains open. The used parameter values are saved on the tcaprev.pph file within the parameter directory and can be restored by using the menu item LOAD PREVIOUSLY USED PARAMETERS.

If VM is the connected host operating system and the VM/CP command tab dialog does not contain a target file name specification for the returned result, the button shows **Display**. In all other cases, the **Execute** button is shown.

Load/Save Parameters

Select this button to display the following menu:

CONTENT	DESCRIPTION
Load previously used parameters	Loads the parameter values, which have been used during the last data transfer operation.
Load parameters	Opens a file selection window. From this window, select a parameter file with the extension .pph. The parameter values stored in this file are set in the appropriate dialog controls.
Save parameters	Opens a window to select a file to store the actual values of the dialog controls. Either a new parameter file may be entered or an existing one may be selected from the list
Set as default	Saves the currently-set dialog values in a specific tcadef.pph file within the parameter directory. If it does exist, the values of this file instead of the program's standards are used when a PC => Host Transfer dialog is started. A window opens and displays the saved standard file name and any user information.
Reset default	Deletes an existing standard file. The program's standards are used when the PC => Host Transfer dialog is started.
List of the most recently used parameter files	The most recently used 4 parameter files will be displayed at the end of the menu. Selecting a line will load the parameter file.

Close

When this button is selected, the dialog closes.

Help

Displays online help for the open dialog.

2.2.3.1.1 PC Tab Dialog

Use this dialog to specify the PC data to be transferred. The related input fields display based on the PC data type. This part of the dialog allows you to specify the PC data to transfer. Depending on the PC data type the related input fields display.

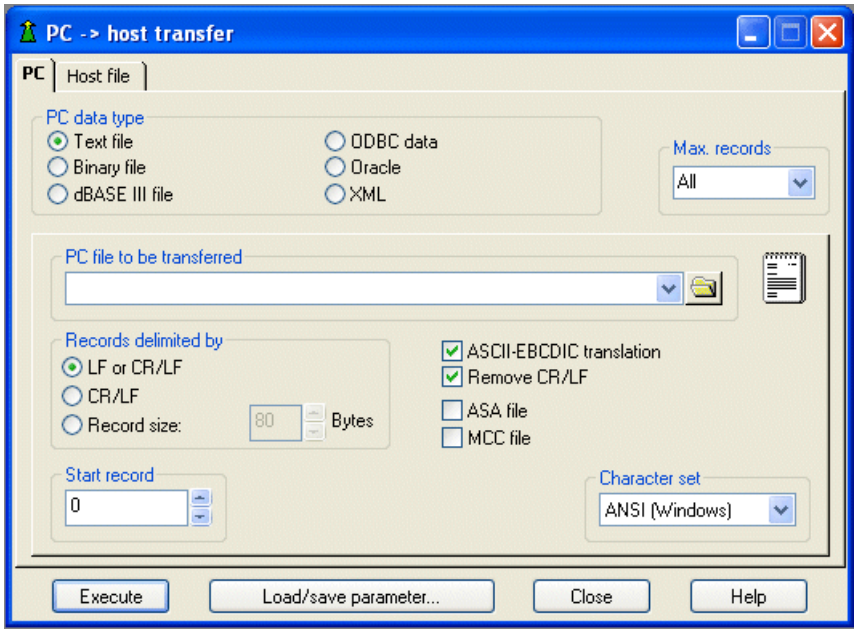


Fig. 57: PC => Host Transfer - Tab Dialog "PC"

The dialog components are described below:

PC data type

This option allows you to select the source data type for a data transfer to the host. You can select from the following:

- Text file
- Binary file
- dBASE III file
- ODBC data
- ORACLE
- XML

If the data is ASCII text, which may be processed on the host in a next step, it is recommended that you use the format *Text* file. It converts all

ASCII characters into EBCDIC code, which is used on the host (see also ASCII/EBCDIC Code Pages, page *201*).

If binary data (programs, packed files, pictures, etc.) is transferred, use the Binary file type. The PC file is sent to the mainframe without conversion.

If the file on the PC is a database in dBASE format, it is possible to convert the fields and data during the transfer. Supported output formats are IBM IXF (e.g. for QMS) and structured EBCDIC text.
If the processed data is accessed via the standard database interface ODBC, the select ODBC Data Most Windows versions ('95, '98, NT) contain several ODBC drivers (e.g., for data in MS Access and SQL Server format). Additionally, these types of modules are often provided with a database system or front-end program (e.g., Oracle).

PC datatype "Oracle" allows a direct access to ORACLE. The OCI-interface (Oracle Call Interface) is being used. The modules needed for this interface are part of the installation of an "Oracle-Frontend". Currently supported is the access to Oracle-Versions Oracle9i and Oracle10g and their corresponding OCI modules.

PC datatype "XML" allows to access XML-file and to transfer the data that are containt in the file to the mainframe. Optionally a validation of the XML file against an XML schema is supported.

Max. records

If this field does not contain the entry **all**, only the defined number of records are transmitted to the mainframe.

2.2.3.1.1.1 Parameters for the PC data type "Text file"

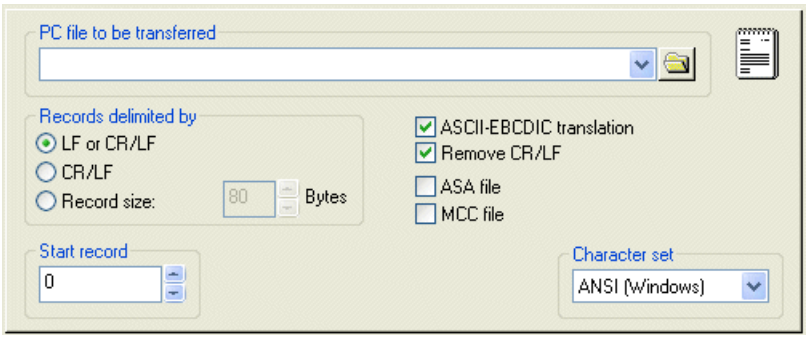


Fig. 58: PC => Host Transfer - Parameters for the PC data type "Text file"

PC file to be transferred

The specified PC file transfers the data to the mainframe. A list of the previously used names displays after selecting the combo box button. If the **Search** button is selected, a dialog appears allowing you to browse the directory structure.

Records delimited by

Use one of the following options to define a record:

- Records delimited by carriage return, line feed or carriage return and line feed
- Records delimited by a given fixed record length.

Each line of a PC ASCII text internally ends with a carriage return / line feed combination (CR/LF). Some text uses only a single carriage return or a single line feed character as line delimeter.
If the transferring file has a fixed record length, a correct record recognition can be granted by specifying the second record limitation option and entering the record length in bytes into the appropriate input field.

ASCII-EBCDIC translation

If this option is active, PC file ASCII characters are translated into EBCDIC characters before they are transferred to the host.

Remove CR/LF

This button is only available with RECORD DELIMITED by CR, LF or CR/LF. If it is active, the CR/LF sequences at the end are removed from the text lines that are transferred.
If this option is not used, tcACCESS transfers and translates these special characters in the same way as any other character. This is useful for SWIFT files.

ASA file

This option is only used for the transfer of a file into a JES or POWER queue. If this option is selected, tcACCESS expects an ASA control character at the first position of each record.

MCC file

This option is only used to transfer a file to a POWER queue. If this option is selected, MCC control characters are written on the host side. The PC file must have an EBCDIC format and each line must start with an MCC character. Additionally, translation must not be used and the ASA option must be active. The record limitation of the PC file can be determined by CR/LF or by using a fixed record length.

Start record

You can transfer the file starting not with the first byte but with a certain record. The first record starts with the number '0'
A record definition can be entered in the Records delimited by group.

Character set

The data transfer between PC and host is always performed in OEM character set. If the ANSI character set is selected, tcACCESS converts the data to be sent and the received data at the PC. If special characters (e.g. ä, ß, ¦,) are transmitted incorrectly, it may result from an incorrectly selected character set.

2.2.3.1.1.2 Parameters for the PC data type "Binary file"

Fig. 59: PC => Host Transfer - Parameters for the PC data type "Binary file"

PC file to be transferred

The specified PC file transfers the data to the mainframe. A list of the previously used names displays after you select the combo box button. When you click the **Search** button, a dialog box displays and allows you to browse through the directory.

Record size

Binary data are divided into records with a fixed length for transfer to the host.

Start record

You can transfer the file starting not with the first byte but with a certain record. The first record starts with number 0.

2.2.3.1.1.3 Parameters for the PC data type "dBASE III file"

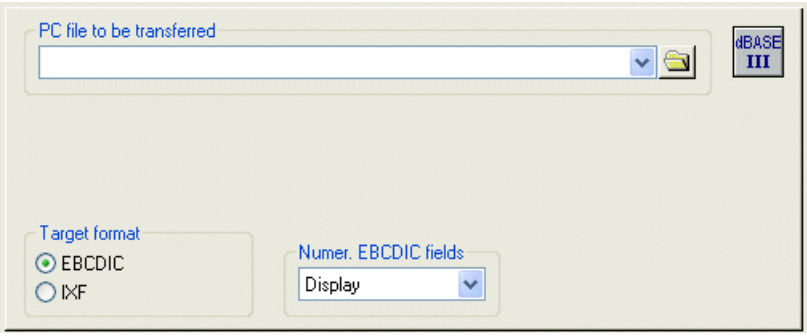


Fig. 60: PC => Host Transfer - Parameters for the PC data type "dBASE III file"

PC file to be transferred

The specified PC file transfers the data to the mainframe. A list of the previously used names displays after you select the combo box button. When you click the **Search** button, a dialog box displays and allows you to browse through the directory.

Target format

dBASE files can be stored on the host as an EBCDIC text file or in a structured IXF format.

Numer. EBCDIC field

If a dBASE file is converted into EBCDIC format on the host, this selection allows you to determine the presentation of numbers in the target.

- Display => Numbers are represented as text.
- Zoned => Numbers are converted into zoned format.
- Packed => Numbers are converted into packed format.

2.2.3.1.1.4 Parameters for the PC data type "ODBC data"

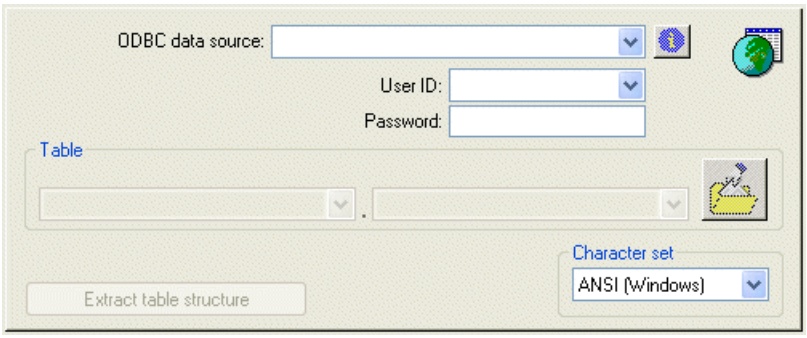


Fig. 61: PC => Host Transfer - Parameters for the PC data type "ODBC data"

ODBC data source

Use the drop down menu next to this field to display a list of all ODBC data sources installed on the PC. To add a data source, use the ODBC Data Sources standard dialog on the Windows Control Panel.

Data source infos

Click this button to open a dialog, which displays information about the selected ODBC data source. It includes the description, parameters, and the conversions between the tcACCESS data types and the internal data types of the ODBC data source.

Table

A table name has either the CREATOR.TABLE format (e.g., MS SQL Server) or TABLE (e.g., MS Access). If the selected ODBC data source uses no creator (owner, etc.), the left input field is not active. Click the button to the right of the field to display all tables and views provided by the ODBC data source. To avoid entering the name of the table, select one of these entries.

Extract table structure

Enter a valid table name for the ODBC data source, then click this button to determine the table structure. The field results are inserted into the Structure definition Tab Dialog, page 122.

Character set

Some ODBC drivers apply the DOS character set (OEM) and not the Windows character set (ANSI). The difference is obvious if special characters are involved (e.g., ü, ä, ß,). If this type of conversion problem arises, you can adjust the character set using this field.

2.2.3.1.1.5 Parameter for the PC datatype "Oracle"

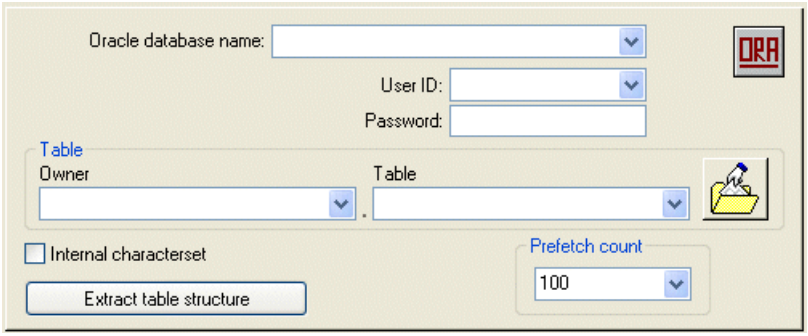


Fig. 62: PC => Host Transfer - Parameter for PC datatype "ORACLE"

Oracle Database name

A "Net Service Name" must be configured on the local machine in order to access an ORACLE database using a logical database name. When an access is performed the "Net Service Name" will be resolved to the required information, i.e. network address or database instance-name. The configuration of the "Net Service Name" can be performed using the ORACLE utility "Net Manager". For further information please refer to the corresponding ORACLE documentation.

Use input field "Oracle database name" to specify the "Net Service Name" that has been configured to access the desired database. As an alternative you may also specify a connection string using the following syntax:

(DESCRIPTION=(ADDRESS=(PROTOCOL=*tcp*) (HOST=*dbhost*) (PORT=*1521*)) (CONNECT_DATA=(SERVICE_NAME=*db1*)))

Starting with databse version Oracle10g the connection string can also look like:

//host:[port][/service name]

The definition of a connection string has the advantage, that a local "Net Service Name" is not required.

User ID

Specify the name of the ORACLE user that should be used to access the database. The specification of a user id is mandatory.

Password

Specify the password for the ORACLE user.

Table

To access an ORACLE table the format of "**CREATOR.TABLE**" must be used. The owner (or creator) of the table must not be identical to the signed-on user. If no owner is specified, the signed-on user will be considered to be the owner of the table. The input field "TABLE" however must be specified.

If the button on the right side is clicked all tables and views (and their synonyms, if available) for the specified user will be displayed. A table entry can either be selected from the list or can be manually entered in the input field.

Delete existing records

With this option active, all currently existing rows in the target table will be deleted. After that, the transfer from the host takes place.

Update existing records

If the target table already contains a record with the same key, this option ensures that the existing record will be updated. If this option has not been set and the record with an identical key already exists in the target database the insert will not take place and the source-record is being skipped.

Internal characterset

During a data-transfer ORACLE always requires the correct characterset at the local client to perform conversions from or to the database characterset.. Normally, the characterset will be obtained from the environment variables "NLS_LANG" and "NLS_NCHAR" or from the corresponding Windows-Registry entries.

For special situations option "Internal characterset" can be specified for a datatranser and the values of "NLS_LANG" and "NLS_NCHAR" will be ignored. If this option is active, the code pages defined with the connection configuration will be used.

IMPPORTANT: If a user defined code page or code page "1160 Thai" is part of the connection configuration, the internal characterset will not be used but instead the values of "NLS_LANG" or "NLS_NCHAR" take effect.

The following assignemt will be used:

Code Page	**Oracle-characterset**
859 – Turkey	TR8MSWIN1254
870 – Easter Europe (Latin-2)	EE8MSWIN1250
875 – Greece	EL8MSWIN1253

420 – Arabic	AR8MSWIN1256
424 – Hebrew	IW8ISO8859P8
1160 – Thai	Values according to "NLS_LANG" or "NLS_NCHAR"
User defined code page	Values according to "NLS_LANG" or "NLS_NCHAR"
All other code pages	WE8MSWIN1252

Extract table structure

Selecting this button will extract the structure of the table specified above. The resulting fields are displayed in the Structure definition Tab Dialog (page 122).

Prefetch Count

Use this parameter to specify the number of records that should be prefetched (read ahead) for every datatransfer request. The intend of this parameter is to increase the performance of the transfer. The allowed values are in the range between 0 (no prefetching) and 999999999, the default is 100.

2.2.3.1.1.6 Parameter for PC-data.format "XML"

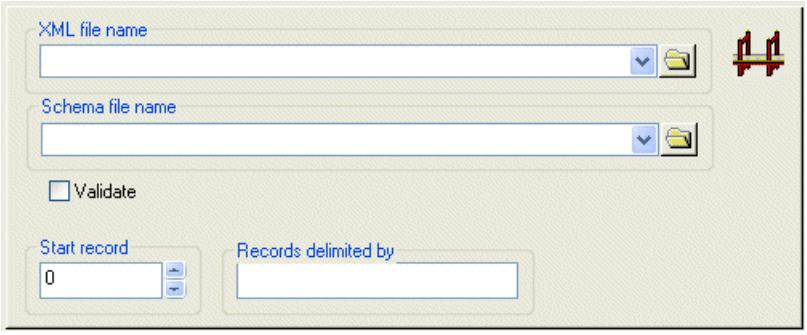


Fig. 63: PC => Host Transfer - Parameter for PC-data-format "XML"

XML file name

The specified XML file will be transferred to the mainframe. Use the listbox to select filenames that have already been used in previous transfers. The BROWSE button can be used to navigate through the file-system to locate the file.

The specified XML file will be processed based upon the supplied structure information (refer to tab "Structure definition") and the supplied record delimiter. The data will be converted based upon the structure definition and transferred as data records.

Schema file name

Optional path to an XML schema file. The schema file can be used to validate the file specified under "XML file name". With option "Validate" active, the validation takes place before the actual transfer of the data. In case that the XML file does not meet the specifications of the schema file, the transfer will be cancelled and a message will be displayed.

The path can either be specified as a local path or as an URI (Uniform Resource Identifier).

Validate

Use this option, if the XML file should be validated based upon the specified schema file prior to transfer.

Start record

Defines the record number where the transfer should be started. The first data record has number "0".

Records delimited by

Use this field to specifiy the XML-tag that delimits the logical record. Do not specify the tab brackets.

A data record consists of XML-tags and -attributes that are imbedded between a start-tag and an end-tag as long as the XML-tags and -attributes are part of the structure definition (the tag itself is defined in the input field). XML-tags and -attributes that are not defined in the structure information are ignored during the transfer. A missing field in a data record that is defined in the structure definition does not result in an error. However it is not allowed that a field/attribute has multiple occurences in a data record.

Note: XML is "case sensitive". Upper- and lower-case characters are recognized.

2.2.3.1.2 Host file Tab Dialog

Depending on the connected host system, different target file types are supported. The following file type displays in this tab dialog:

FILE TYPE	DESCRIPTION
CMS/CP under VM	• Logical files in virtual disks • VM/CMS files • VM/CP print queues • VM/CP reader queues • VM/CP commands
CICS under OS/390	• MVS PDS/PS • JES print queues

FILE TYPE	DESCRIPTION
	• JES reader queues • TD queues • TS queues • Logical files in virtual disks • VSAM files (FCT)

FILE TYPE	DESCRIPTION
CICS under VSE	• TD Queues • TS Queues • Logical files in virtual disks • VSAM files (Batch) • VSAM files (FCT) • VSE Library • VSE Library (Catalog) • SAM • POWER Print queues • POWER Punch queues • POWER Reader queues
TCA/VTAM	• MVS PDS/PS • JES Print queues • JES Reader queues • TS Queues • Logical files in virtual disks • VSAM files (FCT)

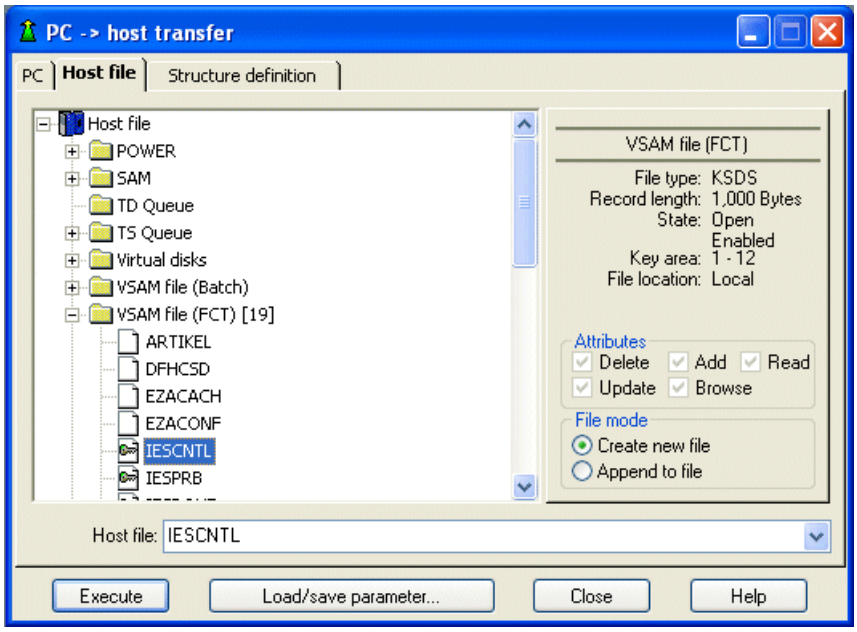


Fig. 64: PC => Host Transfer - Tab Dialog "Host File"

The dialog components are described below:

Tree view

The tree view displays all possible target file types. For some file types, the tree view displays which type currently exists on the mainframe.

If a file in a tree view is selected, the name of the file is displays in the Host file field. Only the file in this field is used for processing.

tcACCESS determines the data that displays for a selected tree view component For instance, a '+' in front of an entry since the program detects that there is no appropriate data available. In any case, the number of sub-entries displays in brackets after expanding a tree node.

To search for a file when using the MVS PDS/PS tree view in an OS/390 environment:

1. Enter the First Level Qualifier in the dialog window.

   This is the part of the name up to the first dot. The name is system-specific and is used to search for existing MVS datasets.

   Example:

   In order to find the SYSTEM.TEST.TMP file, you enter the First Level Qualifier SYSTEM in the dialog box. If the read operation is successful, a second dialog box opens automatically.

2. Enter the First Level Qualifier for MVS PDS/PS in the dialog box.

   Notes:

The MVS PDS/PS dataset may consist of several Members. When you open the dataset, all existing Members are detected and displayed. Each Member type in the tree view can be expanded to display additional data.

When trying to display the member list, tcACCESS recognizes whether the PDS is empty or not. For that reason the original tree symbol () changes to () for a PDS with member or () for a PDS without member.

If an entry on the MVS PDS/PS sub-tree is selected, the input field Disk name (VOLSER) and the tab dialog MVS PDS/PS displays.

Similar processing applies to the tree-view of "VSE-Library (Catalog)". VSE-Libraries can be displayed that have no LIBDEF-definition in the CICS-startup job (JCL. A click on the "+"-sign opens a dialog that asks for the input of a VSAM-file, which contains the desired sub-libraries. If the VSAM-file is not defined in the Master-Catalog the corresponding User-Catalog must be defined. Another tree-view opens that presents the selected VSAM library (i.e. "VSE-Library (VSE.BOS.LIBRARY)") and displays the sublibraries and members. Using this method multiple libraries can be displayed, each one with their own tree-view.

A different processing applies to libraries that are part of the LIBDEF-entry. These libraries will be automatically displayed.

If the VM/CMS File sub-tree is opened while the **Shift** key is pressed, a dialog box opens. Enter the filter conditions (e.g., * EC *). If no key is pressed when it is opened, all available CMS minidisks (File mode, FM) are listed.

When you click an entry on the tree a context menu displays, which allows you to update the content. You can also do this by pressing the F5 key.

If a file should be deleted from the tree-view (i.e. a TS-Queue or a VSAM (FCT) file, the corresponding context menu can be displayed by pressing the SHIFT-key and right-mouse click at the same time or by using the key-combination "SHIFT-DELETE".

For all files with a record format "NC" (section "VSAM-file (Batch)) you can use the context menu item "Display as Library". If the file is a VSE-library ift will be treated as a "VSE-Library (Catalog)" and displayed in a tree-view of its own.

VSE-Library Member have four additional commands in their context menu. A member can be locked or unlocked using the "Lock / Unlock"-command. The "Rename" command can also be invoked by the F2-key. Entry "Info" displays a series of information for the selected member.

Special dispaly of host-data-types in the tree-view:

 VSAM ESDS

- VSAM RRDS
- VSAM KSDS
- VSAM VRRDS
- VSAM Physical Keys
- VSAM SAM
- VSAM Disabled
- Library member type OBJ / PHASE
- Unknown file-type

Host File

The specified host file is used as the target for the transfer. You must use the following syntax:

FILE	PROCEDURE	
MVS PDS	Syntax:	The name must consist of several components separated from each other with a dot, followed by the member name in parentheses.
	Example:	SYSTEM.TEST.TMP(MEMBER)
MVS PS	Syntax:	The name must consist of several components separated from each other with a dot.
	Example:	SYSTEM.TEST.TMP
VSE Library	Syntax	A name separated by periods in the format of: *Library.Sublibrary.Member.Typ*
	Example	(VSELIB) must be added to the name. BOS.TESTLIB.COPYBOOK.A(VSELIB)
VSE Library (Catalog)	Syntax	A name separated by periods in the format of: *Library.Sublibrary.Member.Typ*
	Example	(VSELIB) must be added to the name. TESTLIB.BIN.PHASE(VSELIB)

FILE	PROCEDURE	
JES Print Queue	Syntax:	LST(JES)
JES Reader Queue	Syntax	RDR(JES)
TS Queue	Syntax	(TS) must be added to the queue ID.
	Example:	MYQUEUE(TS)
TD Queue	Syntax	(TD) must be added to the queue ID.
	Example:	MYQUEUE(TD)

FILE	PROCEDURE	
Logical files in virtual disks	Syntax	The name of the virtual disk is separated from the name of the logical file with a colon. Directories may be separated with "/" or "\".
	Examples:	TCAVDI.SYSDSK:/TABLES/TEST.FTB (virtual disk in VSAM file) TCAVDSK VDISK D:/TABLES/TEST.FTB (virtual disk in CMS file)
VM/CMS	Syntax	Add (CMS) to the CMS file name composed of file name, file type and file mode (minidisk).
	Example:	CPLINK EC Y(CMS)
VM/CP Print Queue	Syntax	LST(CP)
VM/CP Reader Queue	Syntax	RDR(CP)
VSAM file (Batch)	Syntax	Add (BVSAM) to the VSAM file name
	Example:	DATA12(BVSAM)
VSAM file (FCT)	Syntax	neither dots nor parentheses
	Example:	TCASTAT
SAM	Syntax	Add (SAM) to the file name
	Example	"VSE.DUMP.LIBRARY(SAM)"
POWER Print Queue	Syntax	LST(POWER)
POWER Punch Queue	Syntax	PUN(POWER)
POWER Reader Queue	Syntax	RDR(POWER)

Depending on the host file specified, the following fields and tabs display in the dialog:

FILE	PROCEDURE
MVS PDS/PS	Display of the selection group Disposition for OS/390 write access, input field Disk name (VOLSER) and tab dialog MVS PDS/PS
JES Print Queue	Display of the tab dialog JES Print
VSAM File Batch	Display of fields "Catalog", "Password" and "VSAM SAM". If option "VSAM SAM" has been set, tab "SAM" will be displayed

FILE	PROCEDURE
VSE-Library or VSE-Library (Catalog)	Display of fields "Use VSAM-nam", "Catalog", "Log. Record length" and "data record format" for library member.
SAM	Display of the tab dialog SAM.
VM/CMS File	Display of the selection group Record format
VM/CP Command	Display of the tab dialog VM/CP Command
POWER	Display of the tab dialog POWER.
File Mode	This field determines how tcACCESS should act if the defined host target file already exists. Possible actions are: Create new file: Delete the existing file before the transfer. Append to file: Add the transferred data to the existing file. The Create new mode is only available for the host file types VSAM KSDS, TS Queue, MVS PDS/PS and virtual disks.
Record format	• This selection group is only available in a VM host operating system environment. It is displayed after selecting the VM/CMS File entry of the tree view presentation or by entering a VM/CMS file into the "Host File" field. • If the **Variable** button is selected, the target file on the mainframe is created with a variable record length. If you do not want this, enter the requested value into the input field provided can specify a fixed record length.
Disposition for OS/390 write access	• This selection group is only available in combination with the OS/390 mainframe operating system. It displays after selecting the MVS PDS/PS entry on the tree view presentation or entering an MVS PDS/PS file into the HOST FILE field. • This selection allows you to specify OLD or SHR mode for write accesses on an OS/390 system.
Disk name (VOLSER)	• This input field is only available within the OS/390 host operating system environments. It displays after you select the MVS PDS/PS entry on the tree view or enter an MVS PDS/PS file into the Host File field. • Input into this field allows you to define a disk name (Volume Serial) for non-catalogued MVS PDS/PS files.

Special input fields for data type "VSE-Library" or "VSE-Library (Catalog)"

Use VSAM name

If the VSE-library is not defined via a LIBDEF-entry in the CICS-startup job (JCL) you can activate this check-box and specify the name of the VSAM-file in the input field.

Catalog

If the file specified with "Use VSAM name" is being defined in a "User Catalog" and not in the "Master Catalog", the corresponding "User Catalog" must be specified in field "Catalog".

Record format

The record format of the member can be specified using this selection. Possible values are "Undefined", "Fixed" and "Variable". The logical record length can be specified with type "Fixed".

Logical record length

The logical record length for a file with record format "Fixed".

2.2.3.1.3 Ext. host parameter Tab Dialog

This tab dialog displays when you select the **Display ext. host parameters** button Options dialog (see Options, page *93*).

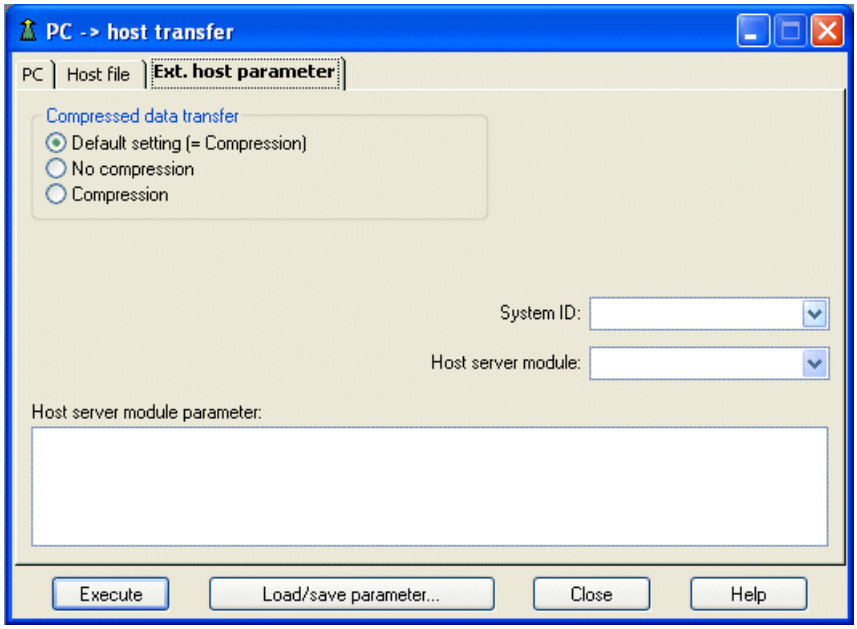


Fig. 65: PC => Host Transfer - Tab Dialog "Ext. host parameters"

The dialog components are described below.

Compressed Data Transfer

Using this option you can activate or deactivate the data compression when transferring data between the PC and the host. A compression might be useful when transferring text data (e.g. query result), it also might be useful when transferring uncompressed binary data. When you chose option "Default setting" the settings defined for the actual slot using the connection configuration will be used.

System ID

Use this field to specify the ID on the host target system for processing. The default is the system with the Host Communication Handler connected to the PC.

This field is not available for mainframes with a VM operating system environment.

Host server module

Use this field to specify the host server module that receives the request. If the field is empty, the system determines the appropriate host server module automatically, depending on the specified host file.

Host server module parameters

Use this field, if necessary, to call a non-standard function of a host server module.

2.2.3.1.4 MVS PDS/PS Tab Dialog

This tab dialog displays the input into the Host File field on the Host File tab dialog and entries selected from the MVS PDS/PS tree view.
It allows you to create a detailed specification of an MVS PDS or MVS PS file.

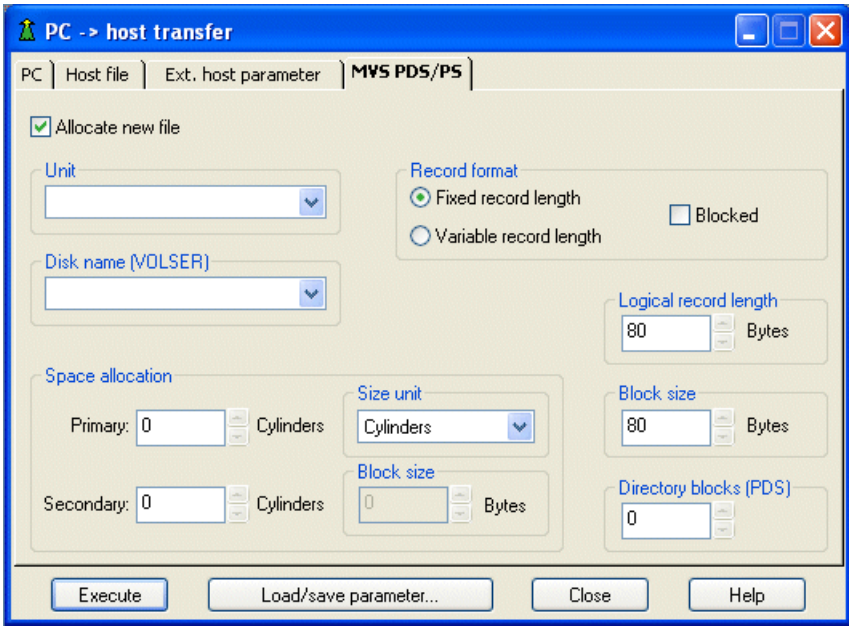


Fig. 66: PC => Host Transfer - Tab Dialog "MVS PDS/PS"

The dialog components are described below.

Allocate new file

Select this option to have all other options (except Disk name (VOLSER)) available. You can determine if you want to create a new file or use an existing one.

Unit

Use this field to enter the system-related UNIT name.

Disk name (VOLSER)

Use this field to define a disk name (Volume Serial) for non-catalogued MVS PDS/PS files or file that is catalogued.

Record format

Specify the record format that is used. The following options are available:

- **Fixed record length, unblocked**
  Requested condition: BLOCK LENGTH is identical with LOGICAL RECORD LENGTH
- **Fixed record length, blocked**
  Requested condition: BLOCK SIZE is a multiple of LOGICAL RECORD LENGTH
- **Variable record length, unblocked**
  Requested condition: BLOCK SIZE is at least the same size as the LOGICAL RECORD LENGTH plus 4
- **Variable record length, blocked**
  Requested condition: BLOCK SIZE is at least the same size as the LOGICAL RECORD LENGTH plus 4

Space allocation

This group allows you to define the primary and secondary space allocation for the file. Select the SIZE UNIT from the drop-down list next to it. Select the BLOCKS UNIT and BLOCK SIZE input field is also activated.

Logical record length

This field contains the record length for the file to be allocated.

Block size

Depending on the selected REORD FORMAT and the LOGICAL RECORD LENGTH, certain limits for the block size are given (see Record format).

Directory blocks (PDS)

If the newly created file becomes a PARTITION DATA SET, use this field to specify the number of blocks for the Member Directory.

2.2.3.1.5 JES Print Tab Dialog

This tab dialog only displays when the specified Host File is recognized as JES Print Queue. It allows you to define the settings required for this transfer type.

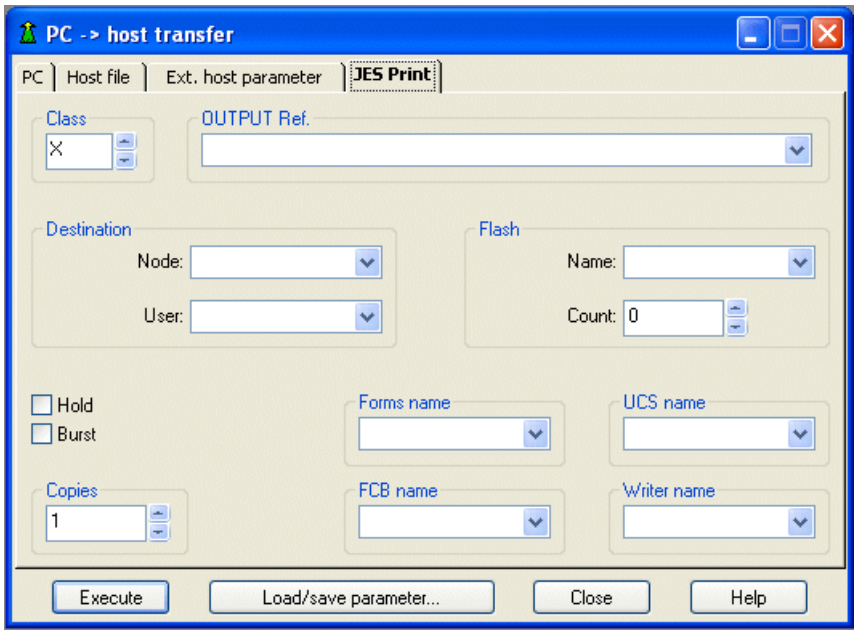


Fig. 67: PC => Host Transfer - Tab Dialog "JES Print"

For a detailed description of the input fields, refer to the JES literature provided by IBM.

2.2.3.1.6 Tab "VSAM SAM" (VSE)

Tab "VSAM SAM" is only displayed when selecting a cataloged BATCH VSAM file . Various parameters can be defined to set or overwrite "VSAM managed SAM"-files.

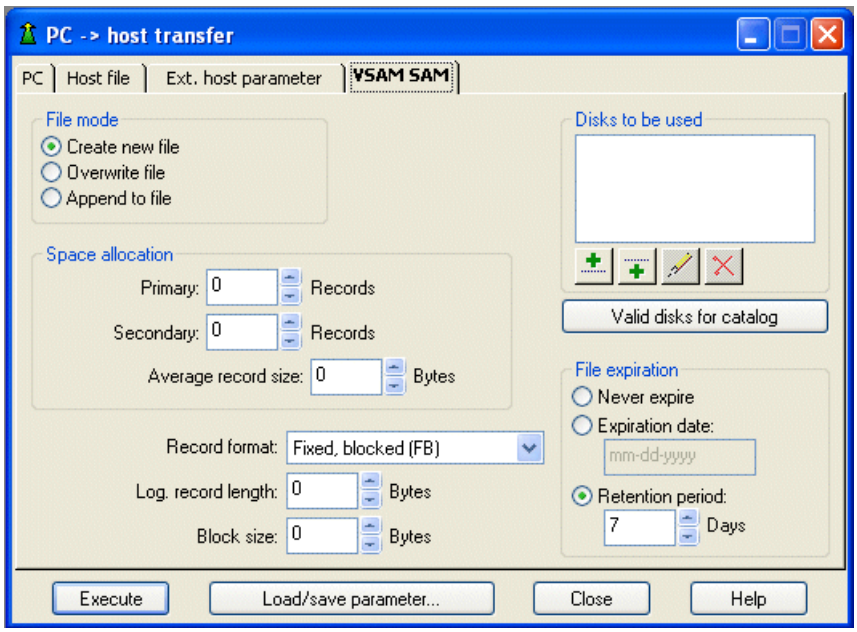


Fig. 68: PC => Host Transfer - Tab Dialog "VSAM SAM"

The dialog components are described below:

File mode

The three input fields of this dialog group can only be used with "Create new file". Primary and secondary space allocations for the the new file can be defined. The definition is done on a record base. The average record length must be defined in the corresponding input field.

Record format

The record format of the target file can be defined. Possible entries are "Undefined", "Fixed", "Fixed blocked", "Variable" and "Variable blocked". The record length can only be specified in conjunction with "Fixed, Blocked".

Log record length

The logical record length for a file with a record format of "Fixed, Blocked".

Block size

Definition of the block size for the new file. If "0" is specified, the maximum possible blocksize will be used.

Disks to be used

The disks, on which a "VSAM managed SAM" file is to be created, are dependent of the catalog, to which the file should belong to. Using the displöayed icons, you can define a list of volumes, which the system should consider to store the host file.

The icons are described below:

 Insert a new entry in front of the selected entry

 Append a new entry after the selected entry

 Edit the selected entry

 Delete the selected entry

Valid disks for catalog

By pressing this button, all volumes will be determined, which are available to the actual catalog.

File expiration

VSAM SAM-files have an expiration date which can be configured. If this date has been expired, the file can be delete after operator intervention. Define either a absolute date or the number of days until the file expires.

2.2.3.1.7 TAB "SAM"

Using this Tab, parameter can be defined to create or overwrite a SAM file (Sequential Access Method).

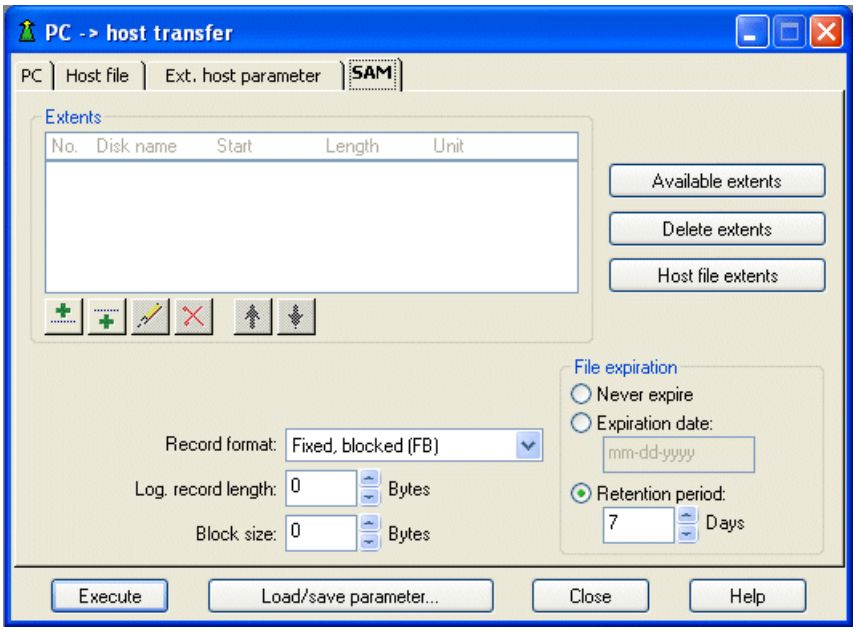


Fig. 69: PC => Host Transfer - Tab Dialog "SAM"

Extents

SAM-files can reside on multiple volume areas or even on multiple volumes. These so called "Extents" must be fully specified and in the correct sequence. You can use the icon to do this. The total length of all extents in bytes is displayed.

The icons are described below:

 Add a new entry in front of the selected entry

 Append a new entry after the selected entry

 Edit a selected entry

 Delete a selected entry

 Move the selected entry one position up

 Move the selected entry one position down

Available extents

When clicking this button, all free disk extents are addedd to the list. If multiple volume types exist, a dialog will be displayed, where you can specify the type. This is necessary, because the extents can only reside on the same volume types.

Delete extents

After a confimation box the listet extents will be deleted at the host.

Host file extents

If a sequential host file has been selected, pressing this button will automatically retrieve all extents for that file.

Record format

The record format of the target file can be defined. Possible entries are "Undefined", "Fixed", "Fixed blocked", "Variable" and "Variable blocked". The record length can only be specified in conjunction with "Fixed, Blocked".

Log record length

The logical record length for a file with a record format of "Fixed, Blocked".

Block size

Definition of the block size for the new file. If "0" is specified, the maximum possible blocksize will be used.

File expiration

VSAM SAM-files have an expiration date which can be configured. If this date has been expired, the file can be delete after operator intervention. Define either a absolute date or the number of days until the file expires.

2.2.3.1.8 Tab Dialog POWER

This tab dialog only displays when the specified Host File is recognized as POWER Reader Queue, POWER Print Queue, or POWER Punch Queue. Specify the required settings for this transfer type on this tab dialog.

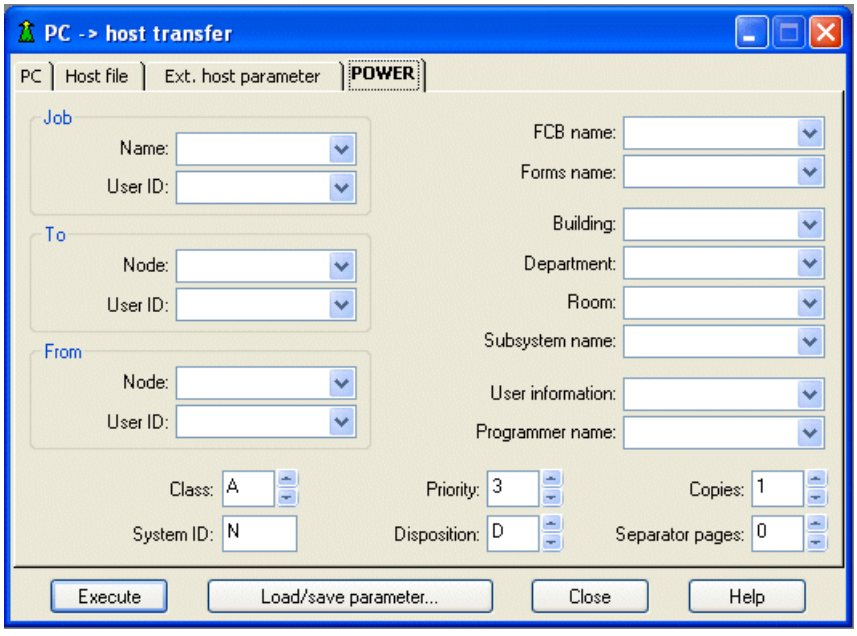


Fig. 70: PC => Host Transfer - Tab Dialog "POWER"

For a detailed description of the input fields, refer to the related POWER literature provided by IBM.

2.2.3.1.9 VM/CP Command Tab Dialog

This tab dialog is displays only if the field with the identical name on the Host File tab dialog contains the input (CP) or if the VM/CP command entry is selected in the tree view display.

It enables the definition of a VM/CP command and, if desired, creates a file on the PC where the query results are saved.

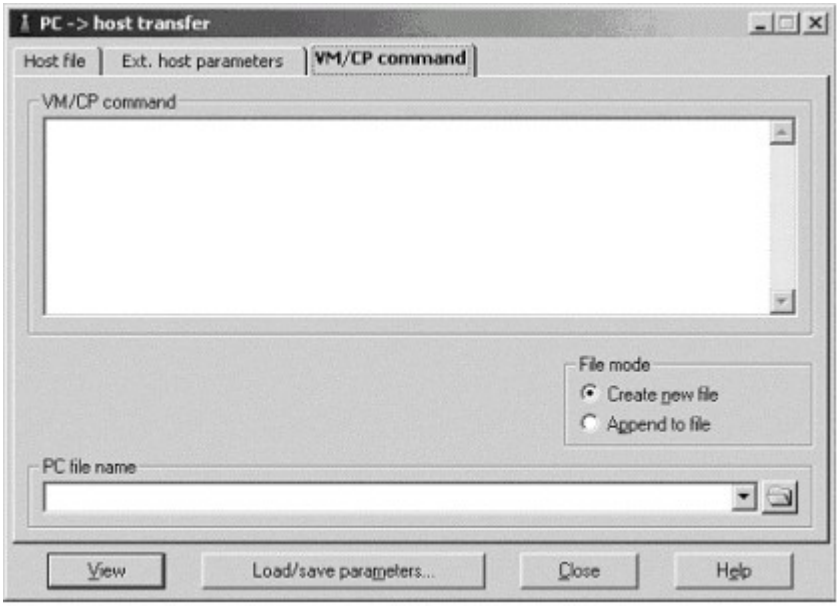


Fig. 71: PC => Host Transfer - Tab Dialog "VM/CP command"

The dialog components are described below:

VM/CP command

The VM/CP command that you enter passes to the mainframe for processing.

File mode

Use this dialog to delete an already existing PC file from the transfer (Create new file) or specify whether the data should be appended to the end of an existing file (Append to file).

PC file name

The specified PC file receives the results of a processed VM/CP command. When you click the **Search** button, a dialog box displays and allows you to browse through the directory.

If this field is empty, the received data is saved in a temporary file and presented for text display using the defined viewing program.

2.2.3.1.10 Structure definition Tab Dialog

This register card only displays for the PC formats ”ODBC”, ”ORACLE” and ”XML”. It allows you to define the target file fields that receive the data. This dialog process is described Structure definition Tab Dialog, page *122*.

2.2.3.1.11 Dialog ”EDIT FIELD DEFINITION“ for PC data format XML

The dialog "Edit field definition" for PC data format XML is different than the other dialogs for PC data formats and is described in this chapter.

Use this dialöog to create or maintain a field definition. It is possible to specify the target host-type and the available fields in the XML file.

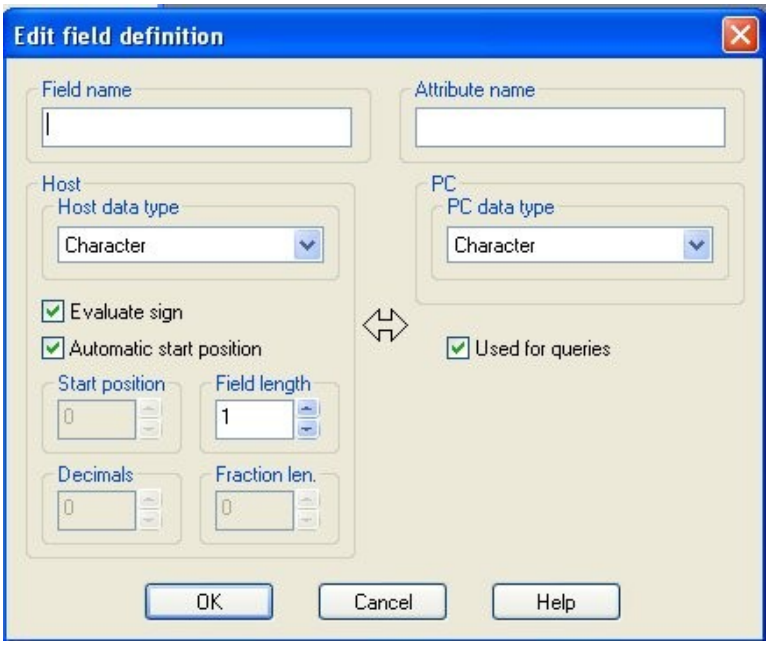


Fig. 72: Dialog ”Edit field definition” for format XML

Field name

Specify the name of an XML-Tag that identifies the defined field in the XML file. The definition of the XML-Tag is without the Tag-brackets abnd is case sensitive. The data between the start and end tag of the defined XML-Tag will be transfered to the host.

If a name has been specified in input field "Attribute name" the attribute values within the XML-tag will be transfered to the host and not the data of the tag.

Attribute name

If the data (i.e. The attribute value) of an XML attribute should be transfered to the host, the corresponding attribute name must be defined of

the field specified in ”Field name”. If an attribute name has been defined, the attribute values and not the data is transfered to the host.

If an XML Tag contains multiple attributes, each attribute can be defined separately as an indivdual field in the structure definition and can be used for the upload.

Used for queries

Specifies whether the field shouuld be used or ignored. If this checkbox is active the corresponding XML data is not transfered to the host.

Host data type

Defines the field type on the host. The following types can be selected:

- **Character**
  EBCDIC character string
  (COBOL PIC X(xx))
- **Binary**
  binary data (i.e. Pictures, sound files)
- **Zoned**
  numeric value in zoned format (variable length)
  (COBOL PIC S9(xx))
- **Packed**
  numeric value in packed format (variable length)
  (COBOL PIC 9(xx) COMP-3)
- **Halfword**
  numeric value in 2 Bytes
  (COBOL PIC 9(4) COMP)
- **Fullword**
  numeric value in 4 Bytes
  (COBOL PIC 9(8) COMP)
- **Floating point**
  floating point number in 8 Bytes

Evaluate sign

Specifies whether a sign (if one exists) should be evaluated.

Automatic start position

If this checkbox is active, tcACCESS automatically calculates the starting position of the field.

Start position

This field can only be used if option ”Automatic start position” is not activated. The offset of this field in the data record can be manually specified.

Field length

The length of the field in the data record can be specified. Based upon the host data type the following rules apply:

- **Character**
  maximum length is ”255“
- **Binary**
  for PC data type ”hexadecimal“ the maximum length is 127, for PC data type ”numeric“ length must be 1 and cannot be edited
- **Zoned**
  not editable because the value is calculated based upon the value specified in ”Decimals”
- **Packed**
  not editable because the value is calculated based upon the value specified in ”Decimals”
- **Halfword**
  not editable because the length is always 2 bytes
- **Ganzwort**
  not editable because the length is always 4 bytes
- **Floating point**
  not editable because the length is always 8 bytes

Decimals

Specifies the number of decimal digits for host data type ”Zoned” and ”Packed”. The field is not used for the other data types.

Fraction len

Specifies the number of fractions for host data types „Zoned“, „Packed“, „Halfword“ and „Fullword“.

PC data type

Specifies the data type of the XML-tag. The mdata must be in a format that corresponds to the "XML-Schema Specification".

- **Charcter**
  ASCII character string
- **Binary**
  refer to PC data type "Hexadecimal"
- **Hexadecimal**
  Representation of the values that should be converted as hexadecimal numbers in a character string (i.e. ”3E4FAA00“)
- **Numeric**
  Representation of the values that should be converted as decimal numbers in a character string
- **Short Integer**

Representation of the values that should be converted as non fraction numbers in a character string

- **Long Integer**
  Representation of the values that should be converted as non fraction numbers in a character string
- **Gleitkomma**
  Representation of the values that should be converted as floating point numbers in a character string
- **Date**
  Date field ("XML Date" Format: '-'? yyyy '-' mm '-' dd zzzzzz?)
- **Time**
  Time field ("XML Time" Format: hh': 'mm': 'ss'. 'sss)
- **Timestamp**
  Field containing date and time ("XML DateTime" Format: '-'? yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss ('.' s+)? (zzzzzz)?

OK

OK validates all input of this dialog. If an error is found a message box displays. If no error has been found the changes will be saved.

Cancel

Terminates the dialog, nothing is saved.

Hilfe

Anzeige der Online-Hilfe für den offenen Dialog.

The following combinations of Host data types to PC data types are possible:

		PC data type									
		Cha-racter	Binar y	Hexa-dec-mal	Nu-meric	Short Inte-ger	Long Inte-ger	Float. point	Date	Time	Time-stamp
Host data type	Character	X			X	X	X	X	X	X	X
	Binary		X	X	X						
	Zoned	X			X	X	X	X			
	Packed	X			X	X	X	X			
	Halfword	X			X	X					
	Fullword	X			X		X				
	Floating point	X			X			X			

2.2.3.1.12 Record selection Tab Dialog

This tab dialog only displays for the PC format ”ODBC” and ”ORACLE”. If you do not want to transfer all data from the data source, this dialog component allows you to define your selection.

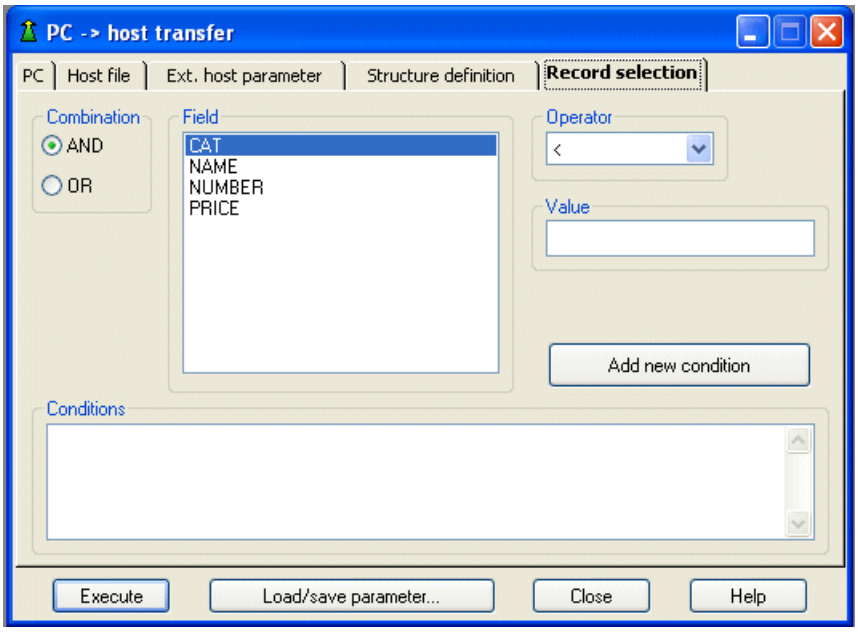


Fig. 73: PC => Host Transfer - Tab Dialog "Record selection"

The dialog components are described below:

Combination

This allows you to select the logical combination of the condition created when you click the **Add new condition** button. It only processes if other entries within the Condition field already exist.

Field

List of all table fields described in the *Structure definition* tab dialog. The USE FOR QUERIES attribute is active on this dialog.

Operator

This is the comparison operator for the Field and Value compbination. The following operators are available:

OPERATOR	DESCRIPTION
<	Less than
<=	Less than or equal to
=	Equal to
<>	Not equal to
>=	Greater or equal to
>	Greater than
LIKE	Matches with a given value

OPERATOR	DESCRIPTION
NOT LIKE	does not match with a given value

Value

The selected field is compared with the content of this input field. Enclose the entry in quotes when you enter characters and binary values. You can enter numeric values with a sign and decimal dot or comma.
If you select LIKE or NOT LIKE as the operator, the Value may have the following meaning:

OPERATOR	DESCRIPTION
_ (space)	Stands for an optional single character
%	Stands for an optional character string For Example: %A CD selects: ABCD YZABCD ABCAXCD but does not select: AB XYZCD AXYCD

Add new condition

Use this function to build a logical term based on the current contents of Combination, Field, Operator, and Value. The system inserts it at the current cursor position at the Condition field.

Condition

Then specified logical expressions are passed as ”WHERE“ - condition to the ODBC-data source or Oracle database.

Examples: Input for the selection of all records Smith or Baker in zone CA:

(NAME = 'Smith' OR NAME = 'Baker') AND ZONE= CA

Input for the selection of all Smith records and in addition all Baker records in zone CA:

NAME = 'Smith' OR NAME = 'Baker' AND ZONE= CA

2.2.3.2 Host => PC Transfer

Open the Host => PC Transfer dialog by selecting the appropriate menu item or by clicking the button as shown below This function allows you to transfer mainframe files to the local PC (download). tcACCESS supports a number of host file types, e.g., VSAM, MVS PDS/PS, TS Queues and Print Queues.

A host session must be started before this menu item available (see Allocate Session, page *89).*

The input fields are initialized internally by the program. However, the start values for the input fields may also be defined individually by pressing the **Load/Save Parameters** button and selecting the SET AS DEFAULT menu entry.

The buttons on this dialog are described as follows:

View / Execute

If the PC tab dialog does not contain a target file name, the **View** button displays and the host file is presented with the program specified in the PROGRAM FOR TEXT DISPLAY field in the Options, page *93*) When the PC file is defined, the button text changes to **Ecute** and the result of the transfer is written into the respective file.

The system checks the validity of the parameters in advance. If an error occurs, a message displays. If all values are correct, the transfer starts and a status window The Data Transfer Status Window, page *269*). When the data transfer is complete, the dialog remains open. The used parameter values are saved in the tcaprev.php file within the parameter directory and can be restored again by using the LOAD PREVIOUSLY USED PARAMETERS menu item.

Load/Save parameters

Select this menu to display the following menu:

MENU ITEM	DESCRIPTION
Load previously used parameters	Loads the parameter values, which have been used during the last data transfer operation.
Load parameters	Opens a file selection window. From this window, select a parameter file with the extension .php. The parameter values stored in this file are set in the appropriate dialog controls. Recovery-files with a file-type of "rhp" can be selected. When loading a Recovery-file an abnormally terminated file-transfer can be restarted.
Save parameters	Opens a window to select a file to store the actual values of the dialog controls. You can enter a new parameter file or select an existing one from the list
Set as default	Saves the currently set dialog values in a specific tcadef.php file within the parameter directory. If it does exist, the values of this file instead of the program's standards are used when a Host => PC Transfer dialog is started. A window opens and displays the saved standard file name and any user information.
Reset default	Deletes an existing standard file. The program's standards are reused when the Host => PC Transfer dialog is started.
Import tcACCESS FTB file	This function allows you to import a Field Table file (extension .ftb), which has either been created for a 2.xx version of tcACCESS or an SQL-Engine table definition file.
List of the most recently used	The most recently used 4 parameter files will be displayed at the end of the menu. Selecting a line will load the

MENU ITEM	DESCRIPTION
parameter files	parameter file.

Close
When this button is selected, the dialog closes.

Help
Displays online help for the open dialog.

2.2.3.2.1 Host file Tab Dialog

Depending on the connected host system, different target file types are supported. The following file type displays in this tab dialog:

FILE TYPE	DESCRIPTION
CMS/CP under VM	• Logical files in virtual disks • VM/CMS files
CICS under OS/390	• DL/I • MVS PDS/PS • TD queues • TS queues • Logical files in virtual disks • VSAM files (FCT)

FILE TYPE	DESCRIPTION
CICS under VSE	• DL/I • TD queues • TS queues • Logical files in virtual disks • VSAM files (Batch) • VSAM files (FCT) • VSE-libraries • VSE libraries (Catalog)
	• POWER print queue • POWER punch queue • POWER reader queue
TCA / VTAM	• DL/I • MVS PDS/PS • TS queues • Logical files in virtual disks • VSAM files (FCT)

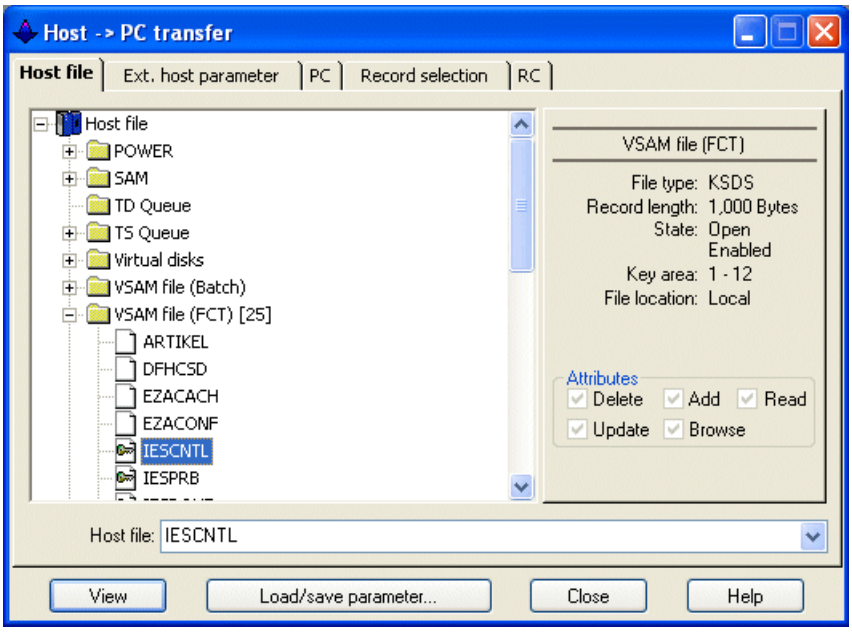


Fig. 74: Host => PC Transfer - Tab Dialog "Host File"

The dialog components are described below:

Tree view

The tree view displays all possible target file types. For some file types, the tree view displays which type currently exists on the mainframe.

If a file in a tree view is selected, the name of the file displays in the HOST FILE field. Only the file in this field is used for processing.

tcACCESS determines the data that displays for a selected tree view component For instance, `+ in front of an entry may be removed since the program detects that there is no appropriate data available. In any case, the number of sub-entries displays in brackets after expanding a tree node.

For instructions on searching for a file when using the MVS PDS/PS tree view in an OS/390 environment, see section PC => Host Transfer, page *97*.

When trying to display the member list, tcACCESS recognizes whether the PDS is empty or not. For that reason the original tree symbol () changes to () for a PDS with member or () for a PDS without member.

If an entry on the MVS PDS/PS sub-tree is selected, the input field Disk name (VOLSER) and the tab dialog MVS PDS/PS displays.
Similar processing applies to the tree-view of "VSE-Library (Catalog)". VSE-Libraries can be displayed that have no LIBDEF-definition in the CICS-startup job (JCL. A click on the "+"-sign opens a dialog that asks for the input of a VSAM-file, which contains the desired sub-libraries. If the VSAM-file is not defined in the Master-Catalog the corresponding User-Catalog must be defined. Another tree-view opens that presents the selected VSAM library (i.e. "VSE-Library (VSE.BOS.LIBRARY)") and displays the sublibraries and members. Using this method multiple libraries can be displayed, each one with a tree-view of their own.

A different processing applies to libraries that are part of the LIBDEF-entry. These libraries will be automatically displayed.

If the VM/CMS File sub-tree is opened while the **Shift** key is pressed, a dialog box opens. Enter the filter conditions (e.g., * EC *). If no key is pressed when it is opened, all available CMS minidisks (File mode, FM) are listed.

When you click an entry on the tree a context menu displays, which allows you to update the content. You can also do this by pressing the F5 key.

If a file should be deleted from the tree-view (i.e. a TS-Queue or a VSAM (FCT) file, the corresponding context menu can be displayed by pressing the SHIFT-key and right-mouse click at the same time or by using the key-combination "SHIFT-DELETE".

For all files with a record format "NC" (section "VSAM-file (Batch)) you can use the context menu item "Display as Library". If the file is a VSE-library it will be treated as a "VSE-Library (Catalog)" and displayed in a tree-view of its own.

VSE-Library Member have four additional commands in their context menu. A member can be locked or unlocked using the "Lock / Unlock"-command. The "Rename" command can also be invoked by the F2-key. Entry "Info" displays a series of information for the selected member.

Special dispaly of host-data-types in the tree-view:

 VSAM ESDS

 VSAM RRDS

 VSAM KSDS

 VSAM VRRDS

 VSAM Physical Keys

 VSAM SAM

 VSAM Disabled

 Library member type OBJ / PHASE

 Unknown file-type

Host File

The specified host file is used as the target for the transfer. The following syntax must be used:

FILE	PROCEDURE	
TS Queue	Syntax:	(TS) must be added to the Queue ID.
	Example:	MYQUEUE(TS)
MVS PDS	Syntax:	The name must consist of several components separated by a dot, followed by the member name in parentheses.
	Example:	SYSTEM.TEST.TMP(MEMBER)
MVS PS	Syntax:	The name must consist of several components separated by a dot.
	Example:	SYSTEM.TEST.TMP
VSE Library	Syntax	A name separated by periods in the format of: *Library.Sublibrary.Member.Typ*
	Example	(VSELIB) must be added to the name. BOS.TESTLIB.COPYBOOK.A(VSELIB)
VSE Library (Catalog)	Syntax	A name separated by periods in the format of: *Library.Sublibrary.Member.Typ*
	Example	(VSELIB) must be added to the name. TESTLIB.BIN.PHASE(VSELIB)

FILE	PROCEDURE	
Logical files in virtual disks	Syntax	The name of the virtual disk is separated from the name of the logical file with a colon. Directories may be separated with "/" or "\".
	Examples:	TCAVDI.SYSDSK:/TABLES/TEST.FTB (virtual disk on VSAM file) TCAVDSK VDISK D:/TABLES/TEST.FTB (virtual disk on CMS file)
VM/CMS	Syntax	(CMS) must be added to the CMS file name composed of file name, file type and file mode (minidisk).
	Example:	CPLINK EXEC Y(CMS)
VSAM-File (Batch)	Syntax	(BVSAM) must be added to the VSAM file name.
	Example:	DATA12(BVSAM)

FILE	PROCEDURE	
VSAM File (FCT)	Syntax	neither dots nor parentheses
	Example:	TCASTAT
SAM	Syntax	(SAM) must be added to the SAM file name.
	Example:	"VSE.DUMP.LIBRARY(SAM)"
POWER Print Queue	Syntax	LST(POWER),JN=job_name,JR=job_userid,CL=class,UI=from_node
POWER Punch Queue	Syntax	PUN(POWER),JN=job_name,JR=job_userid,CL=class,UI=from_node
POWER Reader Queue	Syntax	RDR(POWER),JN=job_name,JR=job_userid,CL=class,UI=from_node

If the defined host file is an MVS PDS/PS type, the additional input **FIELDS** DISPOSITIONS FOR MVS READ ACCESS and DISK NAME (VOLSER) are provided.

FIELD	DESCRIPTION

Disposition for OS/390 read access	• This selection group is only available with the OS/390 mainframe operating system. It is displays after you select the MVS PDS/PS entry on the tree view or enter an MVS PDS/PS file into the HOST FILE field. • This selection allows you to specify OLD or SHR mode for read access on an OS/390 system.
Disk name (VOLSER)	• This input field is only available in the OS/390 host operating system environment. It displays after you select the MVS PDS/PS entry in the tree view by entering an MVS PDS/PS file into the HOST FILE field. • Use this field to define a disk name (Volume Serial) for non-catalogued MVS PDS/PS files.
VSAM File Batch	• Display of fields "Catalog", "Password" and "VSAM SAM". If option "VSAM SAM" has been set, tab "SAM" will be displayed
VSE-Library or VSE-Library (Catalog)	• Display of fields "Use VSAM-nam", "Catalog", "Log. Record length" and "data record format" for library member.
SAM	• Display of tab SAM

Special input fields for file type "VSE POWER Print”.

FIELD	DESCRIPTION
Immediate Commands	By default, no lines will be transferred if they only contain the MCC and no text. By checking the "Immediate Commands” box, these lines will be transferred.

Special input fields for data type "VSE-Library" or "VSE-Library (Catalog)"

Use VSAM name

If the VSE-library is not defined via a LIBDEF-entry in the CICS-startup job (JCL) you can activate this check-box and specify the name of the VSAM-file in the input field..

Catalog

If the file specified with "Use VSAM name" is being define din a "User Catalog" and not in the "Master Catalog", the corresponding "User Catalog" must be specified in field "Catalog".

Record format

The record format of the member can be specified using this selection. Possible values are "Undefined", "Fixed" and "Variable". The logical record length can be specified with type "Fixed".

Logical record length

The logical record length for a file with record format "Fixed".

2.2.3.2.2 Tab Dialog Ext. host parameters

This tab dialog displays when you select the DISPLAY EXT. HOST PARAMETERS Options dialog (see Options, page *93*).

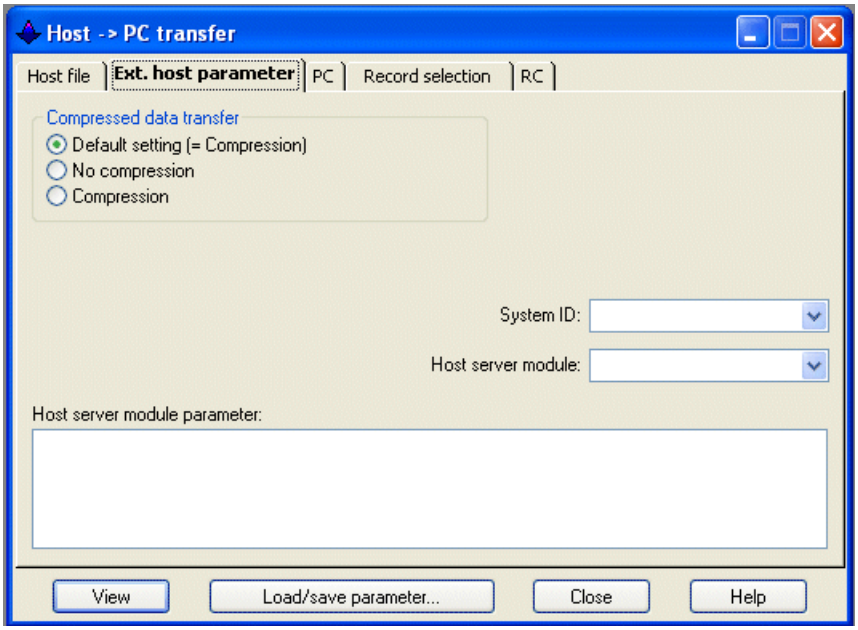


Fig. 75: Host => PC Transfer - Tab Dialog "Ext. Host Parameters"

The dialog components are described below:

Compressed Data Transfer

Using this option you can activate or deactivate the data compression when transferring data between the PC and the host. A compression might be useful when transferring text data (e.g. query result), it also might be useful when transferring uncompressed binary data. When you chose option "Default setting" the settings defined for the actual slot using the connection configuration will be used.

System ID

Use this field to specify the ID on the host target system for processing. The default is the system with the Host Communication Handler connected to the PC.

This field is not available for mainframes with a VM operating system environment.

Host server module

Use this field to specify the host server module that receives the request. If the field is empty, the system determines the appropriate host server module automatically, depending on the specified host file.

Host server module parameters

Use this field, if necessary, to call a non-standard function of a host server module.

2.2.3.2.3 Tab "SAM"

Using this Tab, parameter can be defined to download a SAM file (Sequential Access Method).

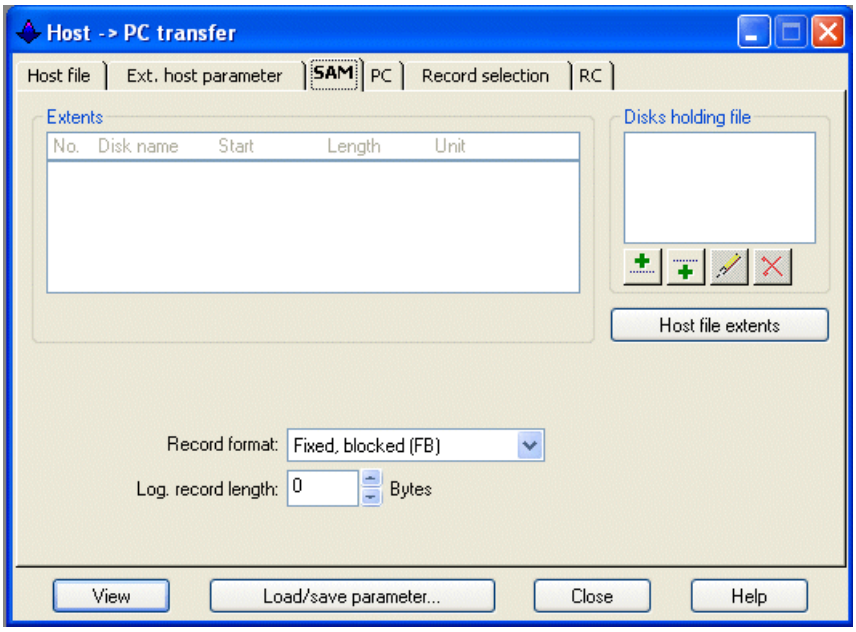


Fig. 76: PC => Host Transfer - Tab Dialog "SAM"

Extents

SAM-files can reside on multiple volume areas or even on multiple volumes. These so called "Extents" must be fully specified and in the correct sequence. They are displayed here for information only.

Disks holding file

When downloading a SAM file, all volumes, which may contain parts of the file must be defined.

The icons are described below:

 Add a new entry in front of the selected entry

 Append a new entry after the selected entry

 Edit a selected entry

 Delete a selected entry

 Move the selected entry one position up

 Move the selected entry one position down

Host file extents

If a sequential host file has been selected, pressing this button will automatically retrieve all extents for that file.

Record format

The record format of the target file can be defined. Possible entries are "Undefined", "Fixed", "Fixed blocked", "Variable" and "Variable blocked". The record length can only be specified in conjunction with "Fixed, Blocked".

Log record length

The logical record length for a file with a record format of "Fixed, Blocked".

2.2.3.2.4 PC Tab Dialog

This tab dialog allows you to specify a target format on the PC. Depending on the PC DATA TYPE the related input fields display.

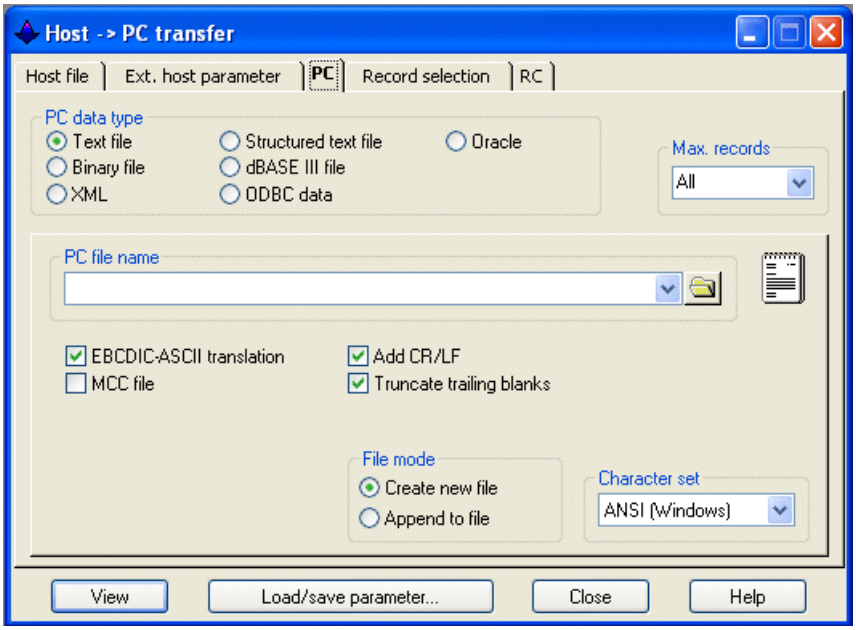


Fig. 77: Host => PC Transfer - Tab Dialog

The dialog components are described below.

PC data type

Select the desired PC data type from the following options:

- Text file
- Binary file
- XML
- Structured text file

- dBASE III file
- ODBC data
- ORACLE

If the transfer data is EBCDIC texts, which should also be processed on a PC, it is recommended that you use the TEXT TRANSLATION option. Since EBCDIC code is used on the host, it converts to ASCII characters (see ASCII/EBCDIC Code Pages, page *201*).

When binary data (programs, packed files, pictures, etc.) are transferred, it is recommended that you select the BINARY FILE option. Then a mainframe file transmits to a PC without any conversion.

The PC data formats Structured text file, dBASE III file, XML, ODBC and ORACLE require a structure definition (see page *122*) of the host source file because they save data on a field level.

If data is stored on a PC via the standard database interface ODBC, you must select ODBC Data. Most of the Windows versions ('95, '98, NT,) contain several ODBC drivers (e.g., for data in MS Access and SQL Server format). Additionally, these types of modules are often supplied with database systems or database front ends (e.g., Oracle).

PC datatype "Oracle" allows a direct access to ORACLE. The OCI-interface (Oracle Call Interface) is being used. The modules needed for this interface are part of the installation of an "Oracle-Frontend". Currently supported is the access to Oracle-Versions Oracle9i and Oracle10g.

Max. records

If this field does not contain **all**, only the specified number of records transfer to the mainframe.

2.2.3.2.4.1 Parameters for the section "Text file"

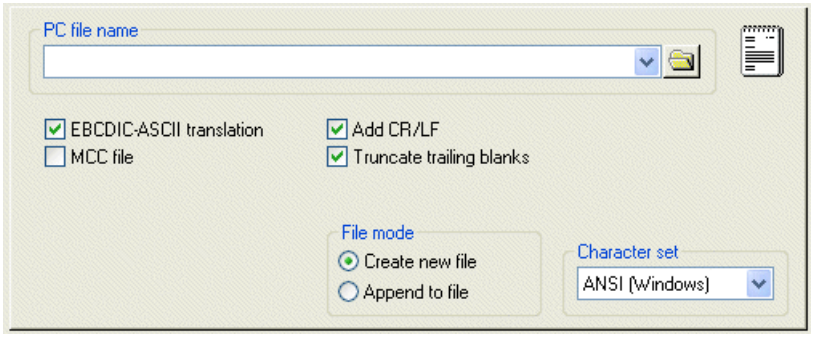


Fig. 78: Host => PC Transfer - Parameters for the PC data type "Text file"

PC file name

The specified PC file is the target for the data transfer from a host. A list of the previously used names is available after clicking the combo box button. When you click the **Search** button, a dialog box displays and allows you to browse through the directory.

EBCDIC-ASCII translation

If this button is active, the EBCDIC characters of the host file are translated into ASCII characters after the transfer.

MCC file

When this parameter is active, each record transferred from the host is prefixed with an MCC control character independent of the selected transfer mode.

Add CR/LF

Using this option adds a carriage return / line feed combination at the end of each transferred record. This option is recommended if host text transfers are lacking line feed and is stored on the PC as a single line of characters.

Truncate trailing blanks

This option removes all blanks at the end of each record (text line).

File mode

There are two options available. If the **Create new file** option is selected, an already existing file is deleted before the transferred data is received. If the **Append to file** option is selected, the received data is added to the existing file.

Character set

The data transfer between PC and host is always performed in OEM character set. If the ANSI character set is selected, tcACCESS converts the data to be sent and the received data at the PC. If special characters (e.g. ä, ß, ¦,) are transmitted incorrectly, it may result from an incorrectly-selected character set.

2.2.3.2.4.2 Parameters for the section "Binary file"

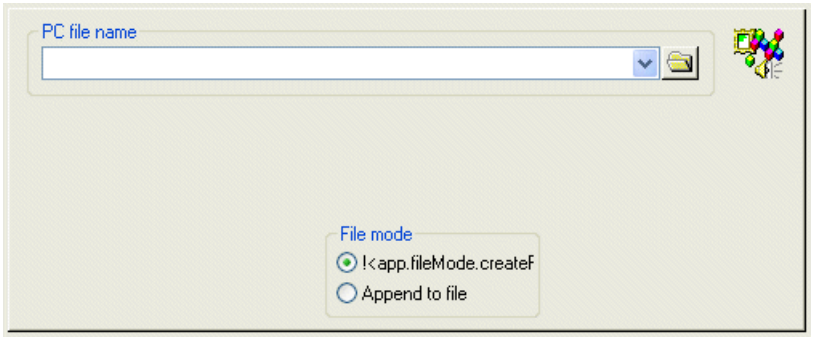


Fig. 79: Host => PC Transfer - Parameters for the PC data type "Binary file"

PC file name

The specified PC file is the target for the data transfer from a host. A list of the previously used names is available after clicking the combo box button. When you click the **Search** button, a dialog box displays and allows you to browse through the directory.

File mode

There are two options available. If the CREATE NEW FILE option is selected, an already existing file is deleted before the transferred data is received. If the APPEND TO FILE option is selected, the received data is added to the existing file.

2.2.3.2.4.3 Parameter for the section "XML"

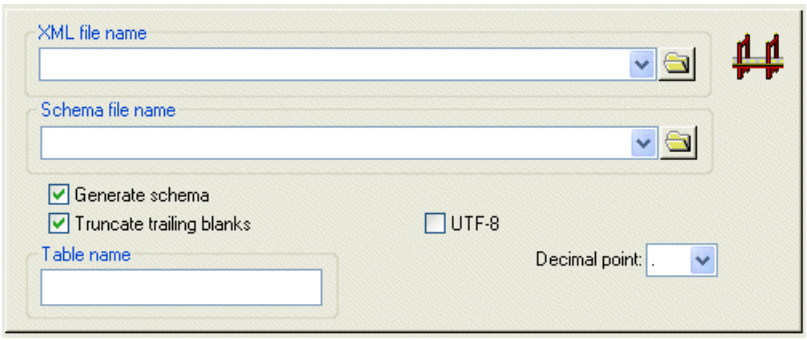


Fig. 80: Host => PC Transfer - Parameter for the PC-file-format "XML"

XML-file name

The XML file is the name of the target file for the file transfer from the mainframe. You can select the file name from the list of previously used file names. You can also browse for the file name by clicking on the corresponding symbol.

If the name of the file has not been specified, the file will be written to a temporary file and will be displayed using the program specified in the Options (page 93).

Schema file name

The name of a target XML-schema file. In addition to the file name the check-box "Write schema file" must be active.. If a schema file has been specified, a link to this file will be written into the XML-instance, even if no XML-schema will be created.

You can select the file name from the list of previously used file names. You can also browse for the file name by clicking on the corresponding symbol.

Write schema file

Activating this option will create a schema file which will define the layout of the created XML-instance. If the name of the file has not been specified the file will be written to a temporary file and will be displayed using the program specified in the Options (page 93).

Truncate trailing blanks

This option remove all trailing blanks from a data record (text line).

Table name

The specified table name will be used as a Root-Tag in the XML-instance. If no name has been defined the name of the host file will be inserted.

A valid XML-name must be used. "Name" and "Tag" must always start with a letter, an underscore or colon followed by "Name Characters". Also, the name must not start with the character string "xml".

With option "UTF-8" active, the table name may also contain UNICODE characters.

Note: The specification of the table name is optional. However, certain host file names may contain invalid characters (i.e. TS-queues), the name always is suffixed by "(TS)"). Brackets are not allowed for "Name" and "Tag" in XML.

UTF-8

The output files (XML-file, Schema-file) are created with UTF-8 coding.

Decimal point

Specify the character that should be used to indicate a decimal point.

Note: The specification for an XML-Schema demands that a period must be used for the decimal point for data-types xsd:decimal, xsd:float and xsd:double. If the XML-file should be validated against an XML-Schema and a comma "," has been used for the decimal point you will receive an error message for the above data-types.

2.2.3.2.4.4 Parameters for the section "Structured text file"

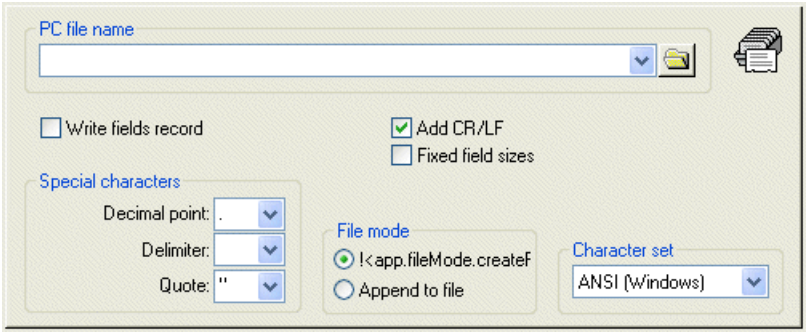


Fig. 81: Host => PC Transfer - Parameters for the PC data type "Structured text file"

PC file name

The specified PC file is the target for the data transfer from a host. A list of the previously used names is available after clicking the combo box button. When you click the **Search** button, a dialog box displays and allows you to browse through the directory.

Add CR/LF

This option adds a carriage return / line feed combination at the end of each transferred record.

Fixed field sizes

If this option is active, the length specifications in the structure definition applies to all data fields (table style output). Otherwise, the values are saved optimized, i.e., blanks at the end (character strings) or zeros at the beginning (numbers) are suppressed.

Decimal point

This field allows you to define the character that represents the decimal position on the PC.

Delimiter

The defined character to separates individual fields from each other.

Quote

This character to encloses character string fields.

File mode

There are two options available. If the CREATE NEW FILE option is selected, an already existing file is deleted before the transferred data is received. If the APPEND TO FILE option is selected, the received data is added to the existing file.

Character set

The data transfer between PC and host is always performed in OEM character set. If the ANSI character set is selected, tcACCESS converts the data to be sent and the received data at the PC. If special characters (e.g. ä, ß, ¦,) are transmitted incorrectly, it may result from an incorrectly-selected character set.

2.2.3.2.4.5 Parameters for the section "dBASE III file"

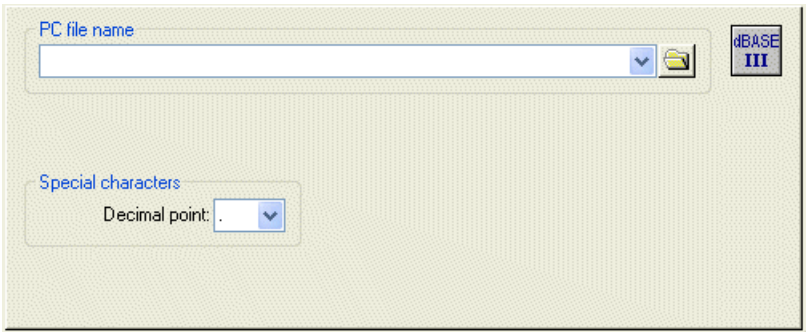


Fig. 82: Host => PC Transfer - Parameters for the PC data type "dBASE III file"

PC file name

The specified PC file is the target for the data transfer from a host. A list of the previously used names is available after clicking the combo box button. When you click the **Search** button, a dialog box displays and allows you to browse through the directory.

Decimal point

This field allows you to define the character that represents the decimal position on the PC.

2.2.3.2.4.6 Parameters for the section "ODBC data"

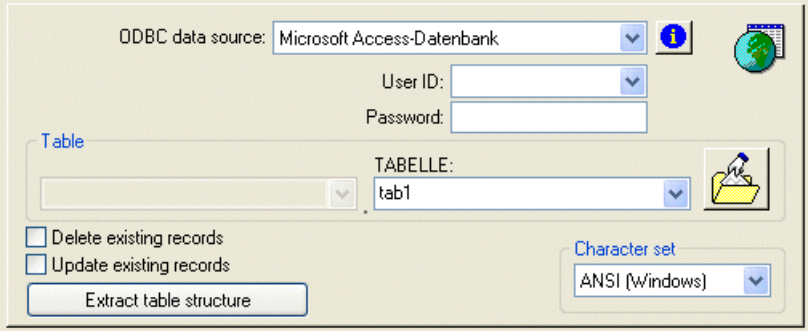


Fig. 83: Host => PC Transfer - Parameters for the PC data type "ODBC data"

ODBC data source

Opening this field displays a list of all ODBC data sources installed on the PC. The ODBC Data Source standard dialog on the Windows Control Panel can be used to add data sources.

Data source infos

Click this button to open a dialog to display information about the selected ODBC data source. The dialog includes the description, parameters, and conversions between the tcACCESS data types and the internal data types of the ODBC data source.

Table

A table name has either the CREATOR.TABLE format (e.g., MS SQL Server) or TABLE only (e.g., MS Access). If the selected ODBC data source has not used a creator (owner, etc.), the left input field is inactive. When you click the right button all available tables and views via the ODBC data source display. You may select the desired table from this list.

Delete existing records

When this option is active, the records in the table will be deleted before the new records will be loaded.

Update existing records

If a **Record already exists** error occurs while inserting data into an ODBC data source and this option is active, it attempts to update the record.

Extract table structure

After a valid table name of the ODBC data source is entered, click this button to determine its structure. The result fields are inserted into the Structure definition Tab Dialog, page *122*.

Character set

The data transfer between PC and host is always performed in OEM character set. If the ANSI character set is selected, tcACCESS converts the data to be sent and the received data at the PC. If special characters (e.g. ä, ß, ¦,) are transmitted incorrectly, it may result from an incorrectly-selected character set.

2.2.3.2.4.7 Parameter für das PC-dataformat "Oracle"

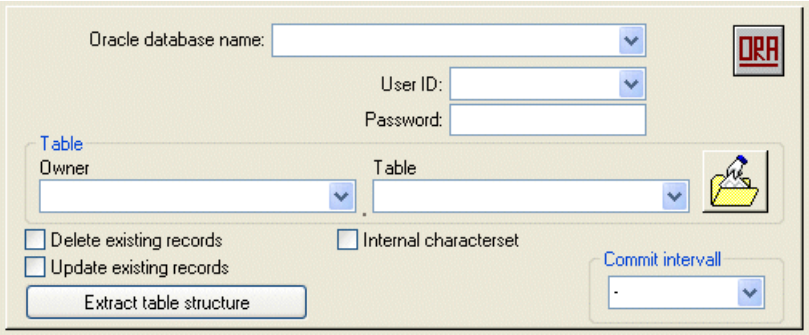


Fig. 84: Host => PC Transfer - Parameter for the PC-dataformat "ORACLE"

Oracle Database name

A "Net Service Name" must be configured on the local machine in order to access an ORACLE database using a logical database name. When an access is performed the "Net Service Name" will be resolved to the required information, i.e. network address or database instance-name. The configuration of the "Net Service Name" can be performed using the ORACLE utility "Net Manager". For further information please refer to the corresponding ORACLE documentation.

Use input field "Oracle database name" to specify the "Net Service Name" that has been configured to access the desired database. As an alternative you may also specify a connection string using the following syntax:

(DESCRIPTION=(ADDRESS=(PROTOCOL=*tcp*) (HOST=*dbhost*) (PORT=*1521*)) (CONNECT_DATA=(SERVICE_NAME=*db1*)))

Starting with databse version Oracle10g the connection string can also look like:

//host:[port][/service name]

The definition of a connection string has the advantage, that a local "Net Service Name" is not required.

User ID

Specify the name of the ORACLE user that should be used to access the database. The specification of a user id is mandatory.

Password

Specify the password for the ORACLE user.

Table

To access an ORACLE table the format of "**CREATOR.TABLE**" must be used. The owner (or creator) of the table must not be identical to the

signed-on user. If no owner is specified, the signed-on user will be considered to be the owner of the table. The input field "TABLE" however must be specified.

If the button on the right side is clicked all tables and views (and their synonyms, if available) for the specified user will be displayed. A table entry can either be selected from the list or can be manually entered in the input field.

Delete existing records

With this option active, all currently existing rows in the target table will be deleted. After that, the transfer from the host takes place.

Update existing records

If the target table already contains a record with the same key, this option ensures that the existing record will be updated. If this option has not been set and the record with an identical key already exists in the target database the insert will not take place and the source-record is being skipped.

Internal characterset

During a data-transfer ORACLE always requires the correct characterset at the local client to perform conversions from or to the database characterset.. Normally, the characterset will be obtained from the environment variables "NLS_LANG" and "NLS_NCHAR" or from the corresponding Windows-Registry entries.

For special situations option "Internal characterset" can be specified for a datatranser (HOST => PC, HOST SQL-Query, PC => HOST) and the values of "NLS_LANG" and "NLS_NCHAR" will be ignored. If this option is active, the code pages defined with the connection configuration will be used. **IMPPORTANT:** If a user defined code page or code page "1160 Thai" is part of the connection configuration, the internal characterset will not be used but instead the values of "NLS_LANG" or "NLS_NCHAR" take effect.

The following assignemt will be used:

Code Page	**Oracle-characterset**
859 – Turkey	TR8MSWIN1254
870 – Easter Europe (Latin-2)	EE8MSWIN1250
875 – Greece	EL8MSWIN1253
420 – Arabic	AR8MSWIN1256
424 – Hebrew	IW8ISO8859P8
1160 – Thai	Values according to "NLS_LANG" or "NLS_NCHAR"
User defined code page	Values according to "NLS_LANG" or "NLS_NCHAR"
All other code pages	WE8MSWIN1252

Commit Intervall

Parameter "Commit Intervall" defines the number of inserted or updated rows after which a COMMIT command will be executed.. The purpose of this parameter is to increase the performance of the datatransfer. The default is that a COMMIT is executed at the end of the transfer. (after all rows have been inserted into the target-table). The default increases the processing speed for relatively small data amounts and allows to completely "ROLLBACK" in case of an error situation.. When transferring larger amounts of data it is recommended to specify a "COMMIT INTERVAL" (i.e. 100), because otherwise you may recognize a declining transfer speed during the transfer..

If "RECOVERY" has been specified, the COMMIT INTERVAL has no meaning and will be grayed out.

Extract table structure

If a valid ORACLE table-name has been specified, this button must be used to extract the table structure. The resulting field list can be processed with tab Structure definition Tab Dialog (on page 146).

2.2.3.2.5 Structure definition Tab Dialog

Structure definition Tab Dialog

This dialog component is only available after the PC data format, Structured Text file, XML file, dBASE III file, ODBC Driver or ORACLE is selected on the *PC tab dialog*. It allows the user to define the structure (fields) of a host file for selective access. Additionally, the identified data may be stored on a PC database (ODBC, dBASE III format, ORACLE) or as a tabular text file or XML file.

A detailed description of the processed field definitions in this tab dialog is found in this section, see *Edit Field Definition Dialog*.

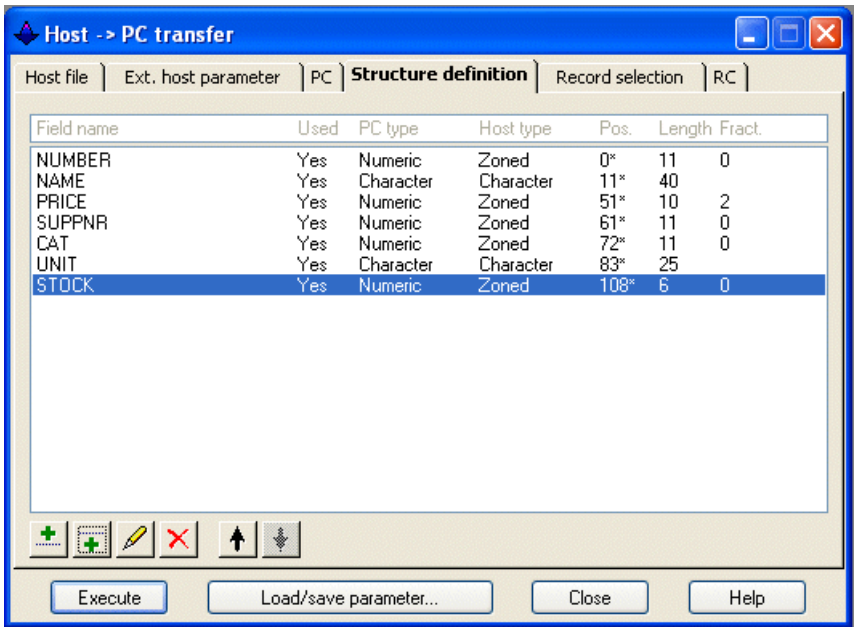


Fig. 85: Host => PC Transfer - Tab Dialog "Structure definition"

The dialog components are described below:

Field definition list

This list is the central element of the tab dialog. It contains all defined fields with their most important criteria. To edit a field, double-click the entry or click the **Edit** icon.

The following table describes the field definitions:

COLUMN NAME	DESCRIPTION
Field name	The table column name, which is used for SQL queries with the SQL-Engine.
Used	This field displays if the field definition is used for queries and transfers.
PC type	Target data type of the field on the PC.
Host type	Source data type of the field on the host.
Pos.	Beginning of the field within a record. A subsequent '*' character shows that the position is calculated automatically based on the preceding field's position and length. The first byte of a record has position '0'.
Length	The field length in bytes.
Fract.	Number of positions behind the decimal point for the host data types Zoned, Packed, Halfword, and Fullword.

The icons are described below:

 Add a new entry in front of the selected entry

 Append a new entry after the selected entry

 Edit the selected entry

 Delete the selected entry

 Move the selected entry one position up

 Move the selected entry one position down

2.2.3.2.6 The Edit field definition dialog

This dialog allows you to create or process a field definition. Use the definition and existing field to describe the desired target file type for host and PC.

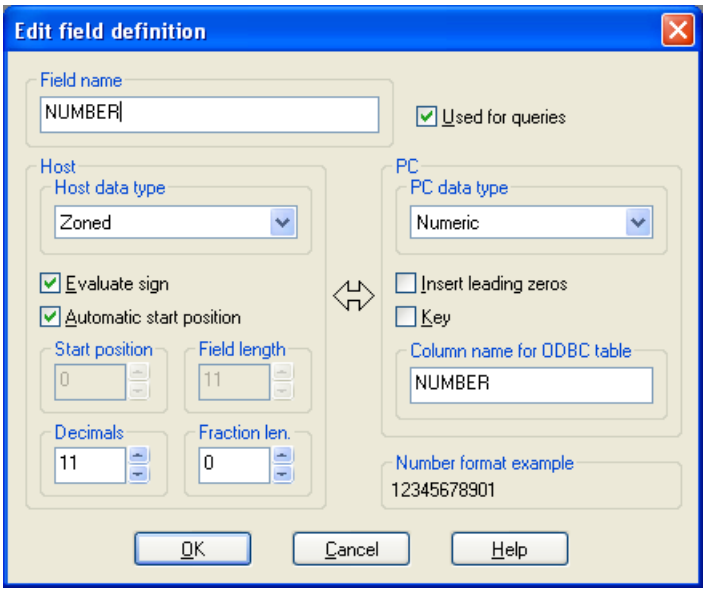


Fig. 86: Host => PC Transfer - Dialog "Edit field definition"

The dialog components are described below:

Field name

A name to identify the defined field. This name is used in the Record selection Tab Dialog (page 152) and with the dBASE III- and XML-file PC data format while writing the target file on the PC. The name must not contain any blanks. It can be up to 20 characters long.

Used for queries

This field determines if a field is to be considered or ignored for the transfer. When this button is inactive, the related data is not processed due to an existing condition nor transmitted from the host.

Host data type

Use this field to specify the field type on the host. The following options are available:

COLUMN NAME	DESCRIPTION
Character	EBCDIC character string (COBOL PIC X(xx))
Binary	Binary data (e.g. pictures, sounds)
Zoned	Numeric value in zoned format (variable length) (COBOL PIC S9(xx))
Packed	Numeric value in packed format (variable length) (COBOL PIC 9(xx) COMP-3)
Halfword	Numeric value (length: 2 bytes) (COBOL PIC 9(4) COMP)

COLUMN NAME	DESCRIPTION
Fullword	Numeric value (length: 4 bytes) (COBOL PIC 9(8) COMP)
Floating point	Floating point number (length: 8 bytes)

Evaluate sign

This option determines if an existing sign should be evaluated.

Automatic start position

If this button is active, tcACCESS automatically calculates the start position of a field due to the position and length of the preceding field.

Start position

This field is only available when the AUTOMATIC START POSITION option is inactive. The offset in the host record may be input manually for this field.

Field length

Use this field to define the length of the field in the host record. Depending on the host data type selected, the following rules must be followed:

COLUMN NAME	DESCRIPTION
Character	The maximum length is 255.
Binary	For the PC data type Hexadecimal, the maximum length is 127, for Numeric, the length must be 1 and cannot be edited.
Zoned	Not editable since the content is calculated from to the value entered in the DECIMALS field.
Packed	Not editable since the content is calculated from the value entered in the DECIMALS field.
Halfword	Not editable. The length for this type of field is always 2 bytes.
Fullword	Not editable. The length for this type of field is always 4 bytes.
Floating point	Not editable. The length for this type of field is always 8 bytes.
Decimals	This value is to determine the number of decimals for the host data types Zoned and Packed. With all other types of data, this field is not considered.
Fraction len.	This value is to determine the number of positions behind the decimal point for the host data types Zoned, Packed, Halfword, and Fullword.

PC data type

The data type to be created on the PC can be selected here.

COLUMN NAME	DESCRIPTION
Character	ASCII character string
Binary	Binary data (e.g. pictures, sounds)
Hexadecimal	Presentation of values to be converted as hexadecimal numbers within a character string (e.g. '3E4FAA00').
Numeric	Presentation of values to be converted as hexadecimal numbers within a character string.
Short integer	Whole numbers stored as a 2-byte-long binary value.
Long integer	Whole numbers stored as a 4-bytelong binary value.
Floating point	8 bytes long, binary floating point number.
Date	DATE field, only for the PC data format ODBC, ORACLE and XML
Time	Time field, only for the PC data format ODBC , ORACLE and XML
Timestamp	Field with date and time, only for the PC data format ODBC, ORACLE and XML

Insert leading zeros

Use this button to determine if free positions at the beginning of values, resulting from Character and Numeric PC data types, should be filled with zeros. If this button is inactive, blanks are used. In case of XML, the leading zeros will be removed and not be replaced by blanks.

Key

To create or update ODBC- or ORACLE-tables, you must provide tcACCESS with information regarding the key fields of a table. This information is automatically determined when you click the **Extract table structure** button, or you can can define it before a new ODBC or ORACLE table is created when transferring data from the host.

An XML schema also contains key information. The specification of key-fields must be made by the user. If the option "Key" has been set for the filed, the field will defined as key-field in the schema-file.

Column name for ODBC table

This name is used to access ODBC- or ORACLE-tables since the FIELD NAME has more restrictions as an ODBC- or ORACLE-column name (e.g., length, blank, special characters). Usually this field contains the same content as the FIELD NAME field. The defined column name will also be used for the creation of XML-files.

OK

When you click OK, all specifications for the field are verified. If there is an error, a message displays. If all values are accepted, the processed field is returned to the requester and the dialog closes.

Cancel

Selecting this button closes the dialog without performing any operation.

Help

Displays online help for the open dialog.
The following host - PC data type combinations are available:

	PC Data Type									
Host Data Type	Char-acter	Binary	Hexa-decimal	Num-eric	Short Integer	Long Integer	Float. point	Date	Time	Time-stamp
Character	X			X	X	X	X	X	X	X
Binary		X	X	X						
Zoned	X			X	X	X	X			
Packed	X			X	X	X	X			
Halfword	X			X	X					
Fullword	X			X		X				
Floating point	X			X			X			

2.2.3.2.7 Record selection Tab Dialog

This dialog component allows you to restrict a desired data transfer to certain criteria:

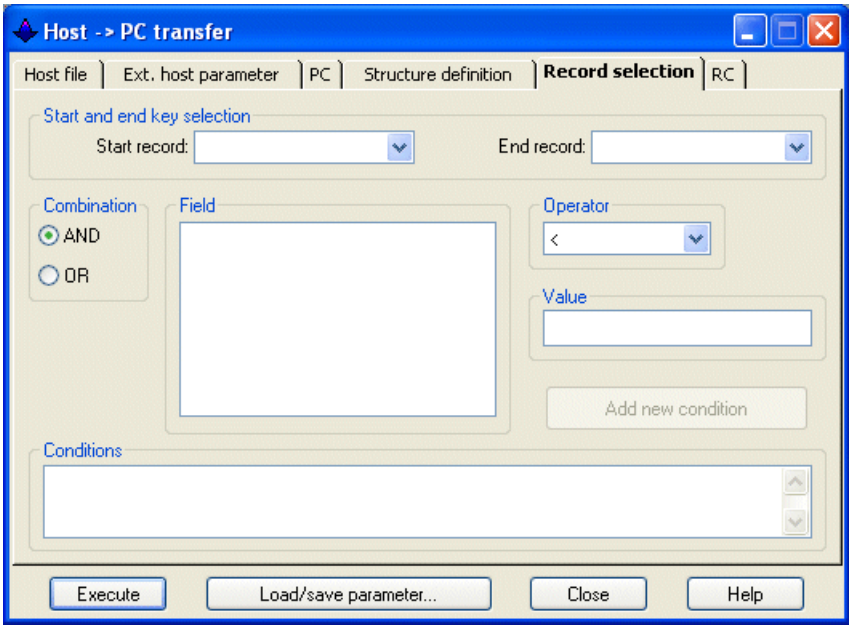


Fig. 87: Host => PC Transfer - Tab Dialog "Record selection"

The dialog components are described below:

Start record, End record

Use these input fields to define the first and the last record that limits the area to be processed on the host.

The valid syntax depends on the type of data used on the host:

FILE TYPE	SYNTAX
Sequential file	VSAM ESDS, VM/CMS, TS Queue, MVS PDS/PS Each record is defined by its record number (first record is number 0).
File with key	VSAM KSDS, DL/I (Its key defines each record.)

Key input for VSAM KSDS:

Text can be directly entered and is automatically translated into EBCDIC. For numeric key values, see the following section entitled *Definition of Numeric Key Values*.

Definition of numeric key values:

Enter numeric data as follows:

%%CData%%

Where C represents the field type, and Data its content.

The following field types are supported:

FIELD TYPE	DESCRIPTION
Hxxx	Hexadecimal number (max. 14 characters)
Px1,x2	Packed, x1 = number of positions, x2 = value (max. 18 characters)
Ixxx	Integer (fullword)
Sxxx	Small integer (halfword)

Examples:

%%P4,-123%%	4 digit packed field	(equivalent to the hexadecimal value '00123D')
%%I033%%	Fullword	(equivalent to the hexadecimal value '00000021')
%%H321%%	2 bytes in hexadecimal format	(equivalent to the hexadecimal value '0321')

Combination

This allows you to select the logical combination of the condition, which is created after clicking the **Add new condition** button. It is processed only if other entries within the CONDITION field already exist.

Determine the processing sequence by using parentheses. Without parentheses, the **AND** combinations are processed first, then the **OR** combinations.

Field

List of all fields described in the *STRUCTURE DEFINITION* Tab Dialog and having the USE FOR QUERIES attribute active.

Operator

Comparison operator for the combination of FIELD and VALUE; the following operators are available:

OPERATOR	DESCRIPTION
<	Less than
<=	Less than or equal to
=	Equal to
<>	Not equal to
>=	greater or equal to
>	greater than
LIKE	Matches with a given value
NOT LIKE	Does not match with a given value

Value

The selected field is compared to the content of this input field.
The selected field is compared with the content of this input field. Enclose the entry in quotes when you enter characters and binary values. You can enter numeric values with a sign and decimal dot or comma.

If you select LIKE or NOT LIKE as the operator, the value may have the following meaning:

OPERATOR	DESCRIPTION
_ (space)	Stands for an optional single character
%	Stands for an optional character string For Example: %A CD selects: ABCD YZABCD ABCAXCD but does not select: AB XYZCD AXYCD

Add new condition

Use this function to build a logical term based on the current contents of COMBINATION, FIELD, OPERATOR, and VALUE. The system inserts it at the current cursor position at the Condition field.

Condition

The conditions you enter are transmitted to the host, which only returns those values to the local PC that match the selected criteria.

Syntax format for conditions:

[NOT] Field [NOT] Operator Value [AND ¦ OR] ...

Note: For a detailed description of the individual components, refer to the explanation of each input field.

Example: Input for the selection of all records ‘Smith’ or 'Baker' in zone CA:

(**NAME = 'Smith' OR NAME = 'Baker') AND ZONE= CA**

Input for the selection of all 'Smith' records and in addition all 'Baker' records in zone CA:

NAME = 'Smith' OR NAME = 'Baker' AND ZONE= CA

It is also possible to enter a field in the following form when a condition relates to a field that is not in the list of fields:

Field type:

C	Characters
B	Binary
P	Packed
Z	Zoned
H	Halfword (2-bytes with sign)
F	Fullword (4-bytes with sign)

Position:

Start position of the field relative to the beginning of the record. (For the first byte position 0 is assumed.).

Length:

Character, Binary:	number of characters
Packed, Zoned:	number of decimals
Halfword:	always 2 bytes
Fullword:	always 4 bytes

Fractions:

Number of decimals after the comma (only for field types halfword, fullword, packed and zoned).

Example:
C(0,20) = 'Miller' OR C(0,20) = 'Meyer' AND P(20,5,0) = 12345

2.2.3.2.8 Tab "RC"

During the data-transfer, situations may emerge that will cancel the transfer (i.e. power outage, network problems etc.). This dialog can be used to restart the data-transfer exactly at that record that has been processed before the abnormal termination. This is of benefit if a larger number of records must be transferred.

The restart of an abnormally terminated data-transfer will be described as follows:

Activate the "Recovery-function" by using the check-box "Enable recovery". Only when this option is active a data-transfer can be restarted. Activating this option must take place BEFORE the actual transfer. During the transfer a recovery file is being created which contains status information of the data being transferred.

After each data-transfer you can find information about the transfer ("Status: Transfer successfully completed" or "Status: Transfer uncomplete"). If the transfer has been successful, the recovery file will be overwritten during the next transfer. "Transfer type" will be "Normal".

In case of an unsuccessful transfer you can load the recovery file (file-type "rhp") like a normal parameter file using the button "Load/Save parameter" and the command "Load parameter...". You must select the file type "Recovery Parameter (*.rhp)" to display the "rhp"-file. After loading the file the original file-transfer parameter will be loaded and additional status information will be processed. "Transfer type" is "Recovery" indicating that, when the file transfer will be startet an actual restart will take place. If the new transfer also fails, you can again load the "Recovery-file" and restart the transfer. The transfer can be repeated as often as necessary in order to complete the transfer.

Note: After a "Recovery-file" has been loaded, the dialog parts "Enable recovery " or "Recovery-file name" are greyed out and cannot be modified. "Transfer

type" is "Recovery". If you decide to not restart the transfer. If you do not want to perform the restart function you can use the "Reset"-button to change the transfer type to "Normal". This will enable new transfers with new settings. The status information will also be reset.

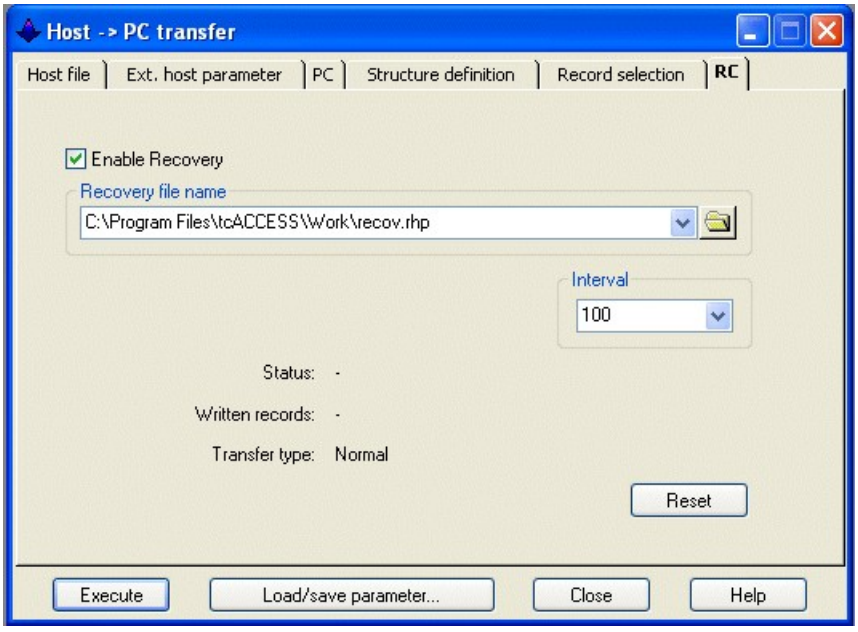


Fig. 88: Host => PC Transfer - Tab "RC"

The dialog components have the following meaning:

Enable recovery

With this option acitve, status information will be written into a "Recovery file" that allows the restart of a data-transfer. Path and file name of the recovery-file (file type "rhp") must be specified.

Recovery file name

Status information about the data transfer will be saved to that file. You can select the file name from the list of previously used file names. You can also browse for the file name by clicking on the corresponding symbol.

To be able to use a recovery file, option "Enable recovery" must be activated before the data-transfer.

Interval

The value specifies, how often a recovery entry is written to the recovery file according to the number of written records of the data transfer. Increasing the value for "Interval" decreases the influence of the recovery on the performance of the data transfer, but if a high value is specified, possibly many records have to be rewritten in case of a recovery transfer. The default value for "Interval" is 100.

Note: **The value of "Interval" has no effect for the PC data type "ODBC data".Status**

If option "Enable recovery" is active, information will be displayed indicating whether the data-transfer was successful or not.

Written records

Specifies the number of records written to the target file or target-table.

Transfer type

"Recovery" will be displayed, when the recofery file has been loaded after an abnormally completed data-transfer. "Normal" will be displayed in all other cases or when the "Reset"-button has been used.

Reset

If a recovery fiule has been loaded to restart a transfer, dialog parts "Enable recovery" and "Recovery file name" are greyed out and cannot be changed. Clicking on the "Execute"-button of the dialog "Host -> PC-Transfer" the data-transfer will be restarted using the settings and status information of the recovery file. If you do not want to restart the transfer you can use the "Reset"-button. The transfer type will be reset to "Normal" and "Enable recovery" and "Recovery file name" can be modified. All status information displayed under tab "RC"will be reset.

2.2.3.3 Host SQL Query

You can open the Host SQL Query dialog Host SQL Q by selecting the appropriate menu item or by clicking the button shown below. This dialog allows you to access mainframe databases. tcACCESS provides a host server module for DB2 and SQL/DS (or DB2 for VSE or DB2 for VMAll). All other database types are referenced using the tcACCESS SQL-Engine (see also the manual *tcACCESS Host Server*).

This menu item is only available when a host session is active (see Allocate Session, page *89*)
The input fields are initialized internally by the program. However, the start values for the input fields may also be defined individually by pressing the **Load/Save Parameters** button and selecting the menu entry SET AS DEFAULT.

The buttons on this dialog are described as follows:

View / Execute

If no file name for the XML-file has been specified on tab "PC", the description of this button is "View". The result of the query will be displayed using the program specified in the Options (page 93). As soon as a PC-file has been specified the description will change to "Execute" and the query result set will be written to the file specified using all file-related parameter.

If the PC-format "Preview" has been specified the data can be viewed using the program defined in the Options (page 93). Optionally the result can be prepared by adding column names and separator lines between the field contents.
PC-data formats "Structured text file", "dBase III file" and "ODBC data" require the definition of a target file or a target table.

Clicking on the button "View" or "Execute" starts the data-transfer from the host. The syntax of the request will be checked. If any errors are being detected an error message will be displayed. If the query is being executed, a status window will be displayed (refer to Details Display, page 270). After the transfer is completed the dialog remains open. The parameter used will be saved to file "tcaprev.psq" in the parameter directory and can be restored using the menu item "Load previously used parameter".

Load/Save Parameters

For a description of the menu items see section Host => PC Transfer, page *127*.

MENU ITEM	DESCRIPTION
Load parameters	Opens a file selection window. From this window, select a parameter file with the extension .psq. Recovery-files with a file-type of "rsq" can be selected. When loading a Recovery-file an abnormally terminated file-transfer can be restarted.
Set as default	Saves the currently set dialog values in a specific tcadef.psq file within the parameter directory. If it does exist, the values of this file instead of the program's standards are used when a PC => Host Transfer dialog is started.
Reset default	Deletes an existing standard file. The program's standards are reused when the Host SQL Query dialog is started.
Save as SQL-Engine Precompiled Command	Save the specified SQL-statement on the host as SQL-Engine Precompiled Command file ("CTB-format"). This option is only available for the SQL-Engine and is not available for "DB2" or "SQL/DS".
Save as SQL-Engine definition	This menu item allows you to save SQL commands entered on the host in the form of an SQL-Engine definition file. If the SQL-Engine selection is active, it is saved as an SQL-Engine View (.VTB format) that appears like a table in the tree view. If DB2 or SQL/DS is selected, the SQL command is saved in an SQL-Engine table definition (.FTB format or .STP). This allows you to access these databases by using the SQL-Engine, which is necessary for table joins and other operations.
List of the most	The most recently used 4 parameter files will be displayed

MENU ITEM	DESCRIPTION
recently used parameter files	at the end of the menu. Selecting a line will load the parameter file.

2.2.3.3.1 Tab Dialog "Host database"

This tab dialog allows you to browse the SQL-Engine, DB2, or SQL/DS tables presented in tree view form. Also, you can specify an SQL command and define various parameters using this dialog.

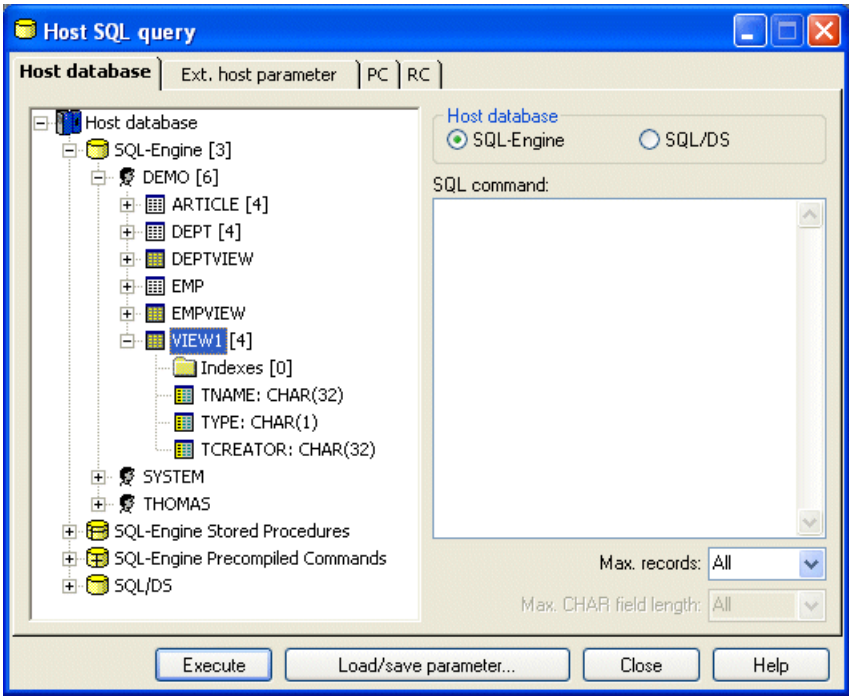


Fig. 89: Host SQL Query - Tab Dialog "Host

The dialog components are described below:

Tree view

The tree view contains the host databases,stored procedures and precompiled commands for tcACCESS SQL Engine, DB2 (OS/390 and SQL/DS (VSE or VM).

When you open a branch of the tree view, all table creators display in the next level (Creator) and displays all of the resulting tables. Tables are classified based on the icon next to it as follows: a real table, a view on a table, a stored procedure or a precompiled command.

Both levels, Creator and Tables, are generated by host access. If a sub-tree within a table is opened, another mainframe access is necessary, whereby the names and data types of all table columns are requested and displayed. For the data types DECIMAL, CHAR, VARCHAR, and LNGVCHAR, positions or length also display in brackets.

The primary keys and indexes of a table entry will be retreived when using the option "Automatic display of table indexes in SQL tree view" (see page 203) or when selecting the table context menu entry "Display indexes". The first subentry "Indexes" contains the primary key and index folders. The table fields belonging to a primary key or index are inserted below the appropiate folder.

The system also inserts a symbol next to a stored procedure. The symbol varies based on the type of parameter. Thus, the symbol → stands for an input parameter, ← represents an output parameter and ↔ an input/output parameter.

An entry for an SQL-Engine Precompiled Command contains the subfolder Input Parameter in addition to the defined fields. All needed input parameter, their data types, length and decimal digits will be displayed.

When you double-click on a table or view entry within the tree or press the space key while the SQL command input field is empty, the latter is filled with the appropriate request (e.g., select * from SYSTEM.SYSTAB). This operation combined with a table column entry on the tree view gives a result (e.g., select TCREATOR from SYSTEM.SYSTAB). If data is already in the SQL command, a complete query is created. The query only processes the table or column name inserted at the current input position within the "SQL command" field. A double-click on a primary key folder, index folder, primary key field or index field builds a SQL command having an appropiate "WHERE" part.

If this operation is processed for a stored procedure in the tree, the related input/output parameters are determined and a query is generated in the SQL command input field. Thereby, only parameters of the type IN or INOUT are applied.

Similar processing applies to Precompiled Commands. Double-clicking the Precompiled command object or pressing the space bar will create the corresponding command in the input field "SQL-command" (i.e.: "execute TEST.COMMAND1 (‚CUSTNAME', NR)"). The input parameter will be displayed as place-holders with their names. The names must be replaced with the parameter values that the EXECUTE should use.

SQL-Engine Table definitions, SQL-Engine Views ‚SQL-Engine Stored Procedures and SQL-Engine Precompiled COmmands can be modified, created or deleted using context menus. These context menus will be displayed when clicking on the corresponding entry in the tree view using the right mouse button.

The administrator options (see Tab Dialog "Menu customization", page *189*) allows you to define the DB2 or SQL/DS tables or views that are considered for the creation of the tree view.

The icons in the tree view are described below:

 Database (SQL-Engine, DB2, SQL/DS)

 Table or Procedure Creator

 Table

 View

 Primary key folder; containing the key table fields

 Index folder; containing the index table fields

 Table field, which is part of a primary key

 Table field, which is part of an index (performance enhancement for queries)

 Table field not being part of any primary key or index

 SQL-Engine Stored Procedures

 SQL-Engine Stored Procedure

 Parameter of a SQL-Engine Stored Procedure

 SQL-Engine Precompiled Commands

 SQL-Engine Precompiled Command

 Input-Parameter of an SQL-Engine Precompiled Command

Host database

The selected database system is accessed for processing the SQL command. If double-clicking on a table on the tree view created it, all settings are correctly defined.

SQL command

Depending on the selected database system, you can input different SQL commands in this field:

DB2

You can enter several SQL commands separated by a semicolon. They are processed as LUW, i.e., either all commands or none of them executes when an error occurs.
The last SQL command can only can be a query command (SELECT ...). After the first query command, all SQL commands are ignored.

The following entries are supported for DB2:

Standard commands such as '**SELECT** ...' or '**UPDATE** ...'

Comments marked with two hyphens at the beginning. (e.g.: '-- **This is a comment**;')

'**COMMIT**'

Confirms the current LUW and activates the auto-commit mode

'**ROLLBACK**'

Terminates the current LUW, reverses the modifications of data effected in this LUW and activates the auto-commit mode.

'NOCOMMIT'

Deactivates the auto commit mode. The client application must perform a ROLLBACK or COMMIT in a subsequent SQL command in order to close the LUW. tcACCESS automatically performs a ROLLBACK when either a non-SQL function has been started or a host session has been ended normally or abnormally.

Note: After a NOCOMMIT command, SQL commands are processed with ISOLATION LEVEL(RR) instead of the standard ISOLATION LEVEL(CS), i.e., before the end of the LUW, it is impossible for other users to alter data that are retrieved with a SELECT command.

Stored Procedures

DB2 Stored Procedures are called as follows:

CALL procedure (parm1, parm2, ...)

Here, PROCEDURE stands for the name of the procedure as it is defined in the SYSIBM.SYSPROCEDURES.PROCEDURE table.

Additional options, PARM1, PARM2, etc. are equivalent to the parameters defined in the PARMLIST field of the called procedure. Only parameters defined as IN or INOUT within a CALL command are applied. OUT and INOUT parameter values are transferred to the client in the same manner as query results. If parameters of this type do not exist, the CALL command is handled like a non-Query command.

If a DB/2 defined Stored Procedure is called, which opens multiple cursors and retriebves records in multiple recordsets, you can use the key word "**GET_RESULTSET**". Examples, how to use this key word can be found in the "Procedures Manual".

SQL/DS (or DB2 for VSE or DB2 for VM)

It is possible to enter several SQL commands, separated by a semicolon. They are processed as an LUW, i.e. either all commands or none are executed if an error occurs.

Only the last SQL command may be a query command (SELECT ...). All SQL commands following the first query command are ignored.

The following input is supported for SQL/DS:

Standard commands such as **'SELECT ...'** or **'UPDATE ...'**

Comments marked with two hyphens at the beginning, e.g**.: '-- This is a comment;'**)

'COMMIT'

Confirms the current LUW and activates the auto-commit mode.

'ROLLBACK'

Terminates the current LUW, reverses the data modifications effected by this LUW, and activates the auto-commit mode.

'NOCOMMIT'

Deactivates the auto commit mode. The client application must perform a ROLLBACK or COMMIT in a subsequent SQL command in order to close the LUW. tcACCESS automatically performs a ROLLBACK when either a non-SQL function has been started or a host session has been ended normally or abnormally.

SQL-Engine

The SQL-Engine allows you to perform queries on non-SQL databases using standard SQL commands. Supported resources are VSAM, DL/I, TS Queues, MVS PDS/PS, DATACOM/DB, ADABAS and others. Joins can be applied on different data sources. Views can be used to present the user exact defined windows on data sources.

The following functions are available:

'SELECT ... FROM ... WHERE ...'
Retrieve data records (inclusive GROUP BY, HAVING, ORDER BY)

'INSERT ... INTO ... VALUES ...'
Insert data records (optionally with subselect)

'UPDATE ...'
Update data records (optionally with subselect)

'DELETE ... FROM ... WHERE ...'
Delete data records

'COMMIT'
Confirm all changes that were made since the beginning of the transaction

'ROLLBACK'
Undo all changes that were made since the beginning of the transaction

'NOCOMMIT'
Deactivate the automatic commit

'CALL'
Call a stored procedure

"EXECUTE"
Execution of a Precompiled Command

The following functions are also supported within expressions:

Individual functions:

BIN(), **DATE()**, **DAYS()**, **DECIMAL()**, **DECODE()**, **HEX()**, **SUBSTR()**, **STR()**, **TIME()**, **TIMESTAMP()**, **USER()**, **CURRENT_ TIMESTAMP()**

Grouping functions:

AVG(), **COUNT()**, **MAX()**, **MIN()**, **SUM(),FIRST(),LAST()**

Operators:

+, **-**, ***, **/**, **||**

Conditional operators:

<, **>**, **=**, **>=**, **<=**, **<>**, **BETWEEN**, **LIKE**, **IN**, **IS NULL**

Logical operators:

AND, **OR**

For a detailed description of the SQL-Engine, see the *tcACCESS Host Server* manual.

IXF file

It is possible to transfer IXF formatted data from an MVS PDS/PS in the same manner as from a database. To do this, the SQL command input field must contain the following command:

IXF=dsn

'dsn' is the valid name of an MVS PDS/PS file.

Max. records

Defines the maximum number of records to be transferred.

Max. CHAR field length

If this field contains a value instead of input 'all' fields of the CHAR type are cut to the defined length. This function allows you to control the volume of data.

2.2.3.3.2 Tab Dialog "Ext. host parameters"

This tab dialog displays when you select the **Display ext. host parameters button** Options dialog (see Options, page *93*).

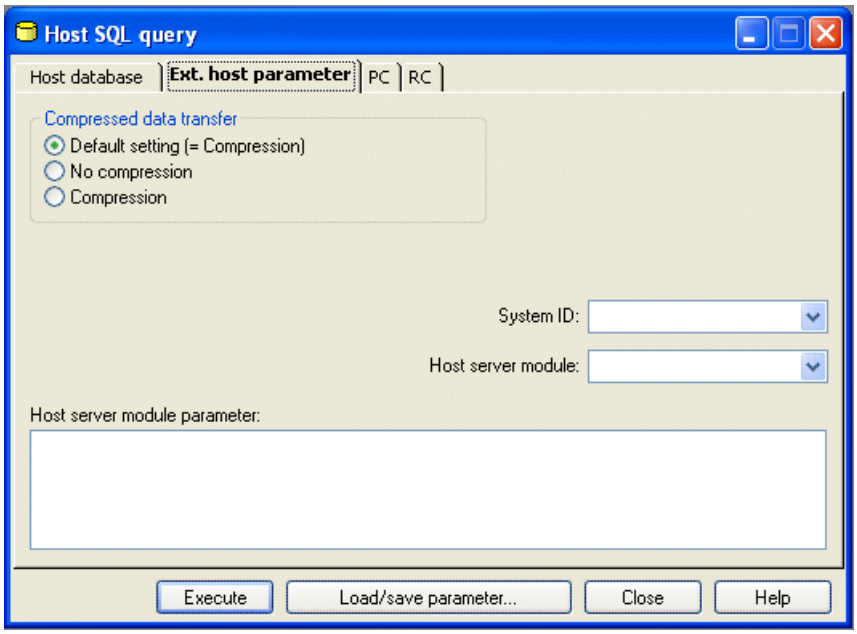


Fig. 90: Host SQL Query - Tab Dialog "Ext. Host Parameters"

The dialog components are described below.

Compressed Data Transfer

Using this option you can activate or deactivate the data compression when transferring data between the PC and the host. A compression might be useful when transferring text data (e.g. query result), it also might be useful when transferring uncompressed binary data. When you chose option "Default setting" the settings defined for the actual slot using the connection configuration will be used.

System ID

Use this field to specify the ID on the host target system for processing. The default is the system with the Host Communication Handler connected to the PC.

This field is not available for mainframes with a VM operating system environment.

Host server module

Use this field to specify the host server module that receives the request. If the field is empty, the system determines the appropriate host server module automatically, depending on the specified host file.

Host server module parameters

Use this field, if necessary, to call a non-standard function of a host server module.

2.2.3.3.3 PC Tab Dialog

This tab dialog allows you to specify a target format on the PC. Depending on the PC DATA TYPE the related input fields display.

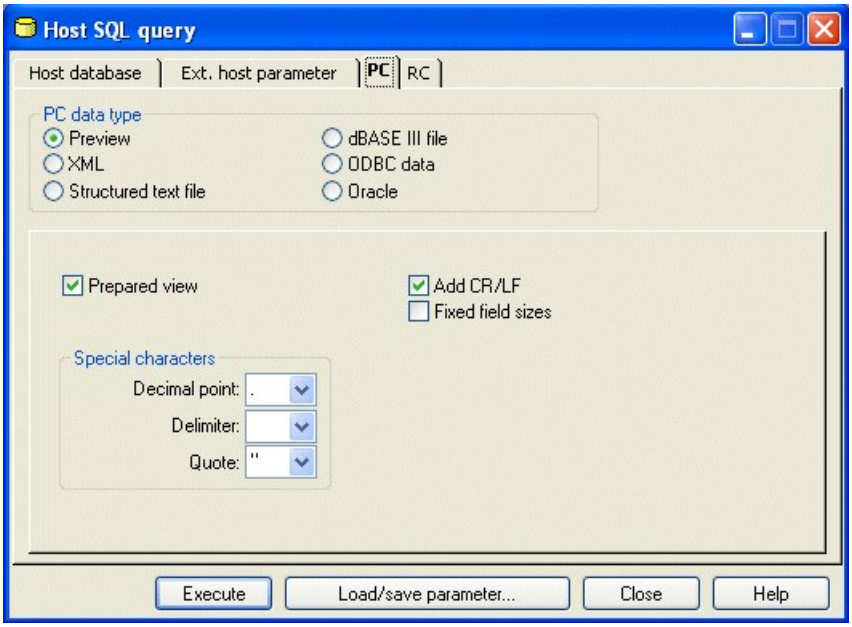


Fig. 91: Host SQL Query - Tab Dialog "PC"

The dialog components are described below:

PC data type

Select the desired PC data type from the following options:

- Preview
- XML
- Structured text file
- dBASE III file
- ODBC data
- ORACLE

If data is stored on a PC using the standard database interface ODBC, you must select ODBC Data. Most of the Windows versions ('95, '98, NT,) contain several ODBC drivers (e.g., for data in MS Access and SQL Server format). Additionally, these types of modules are often supplied with database systems or database front ends (e.g., Oracle).

PC datatype "Oracle" allows a direct access to ORACLE. The OCI-interface (Oracle Call Interface) is being used. The modules needed for this interface are part of the installation of an "Oracle-Frontend". Currently supported is the access to Oracle-Versions Oracle9i and Oracle10g.

If "Preview" has been selected as the PC-data format, the result set will be saved to temporary file and displayed using the program specified in Options (page 93). Optionally the data can be displayed as table with column headings and separation lines. The temporary file will be deleted when tcACCESS is terminated.

2.2.3.3.3.1 Parameter for the "Preview" PC-data format

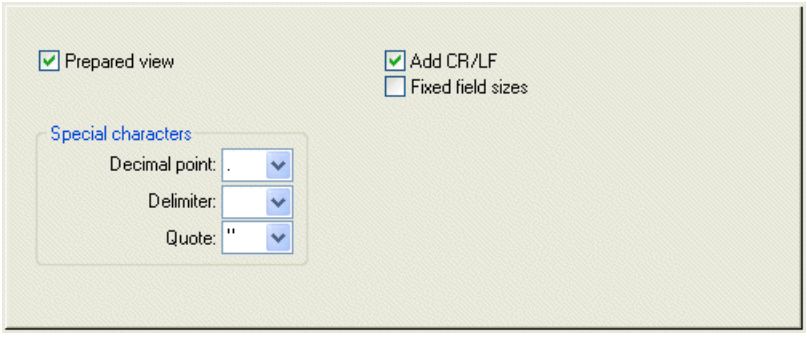


Fig. 92: Host SQL-Query – "Preview" parameter

Prepared view

Activation of this option will display the result data in a table like format. Before the display the data will be written to a temporary file. In addition to the data, column headings using the field names and separation lines will be used to enhance readability. If this option is not activated, the data will be displayed like the file format "Structured text file".

Add CR/LF

A Carriage Return / Line Feed-combination will be inserted at the end of each line (line break).

Fixed field sizes

Using this option, the data will be displayed according to the length designated by the appropriate SQL data-type (tabular display). Otherwise the data will be saved with the minimum display length, i.e.: trailing blanks or leading zeroes will be suppressed.

Decimal point

Specify the character that should be used to indicate a decimal point.

Delimiter

The defined character is used to separate the individual fields from each other.

Quote

This character is used to enclose character string fields.

2.2.3.3.3.2 Parameter for the "XML"-format

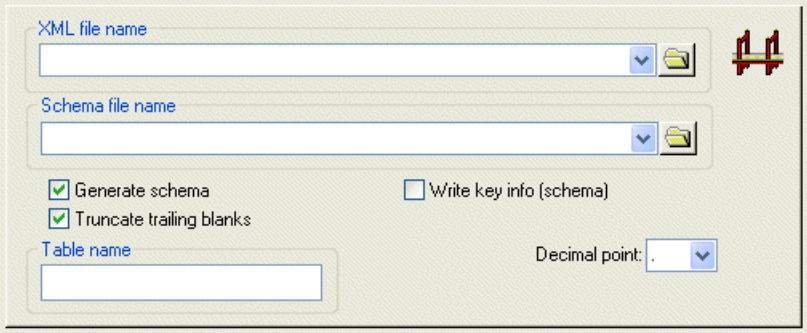


Fig. 93: Host SQL-Query - Parameter for the "XML"-format

XML-file name

The XML file is the name of the target file for the filet-ransfer from the mainframe. You can select the file name from the list of previously used file names. You can also browse for the file name by clicking on the corresponding symbol.

If the name of the file has not been specified the file will be written to a temporary file and will be displayed using the program specified in the Options (page 93).

Schema file name

The name of a target XML-schema file. In addition to the file name the check-box "Write schema file" must be active.. If a schema file has been specified, a link to this file will be written into the XML-instance, even if no XML-schema will be created.

You can select the file name from the list of previously used file names. You can also browse for the file name by clicking on the corresponding symbol.

Write schema file

Activating this option will create a schema file, which will define the layout of the created XML-instance. If the name of the file has not been specified the file will be written to a temporary file and will be displayed using the program specified in the Options (page 93).

Truncate trailing blanks

This option removes all trailing blanks from a data record (text line).

Table name

The specified table name will be used as a Root-Tag in the XML-instance. If no name has been defined the name of the host file will be inserted.

A valid XML-name must be used. "Name" and "Tag" must always start with a letter, an underscore or colon followed by "Name Characters". Also, the name must not start with the character string "xml".

Note: The specification of the table name is optional. However, certain host file names may contain invalid characters (i.e. TS-queues), the name always is suffixed by "(TS)"). Brackets are not allowed for "Name" and "Tag" in XML.

Write key info (schema)

Key- and index-information of the host resource will be evaluated and become part of the XML-schema. If this option is not active, this information will be missing from the XML-schema.

The evaluation of the key- and index-information with an SQL-query can only be performed if data from only one table will be queried. If multiple tables are being referenced in an SQL-statement the evaluation will not be possible, even if this option has been acivated. The same applies to Precompiled Commands. However, you can create an SQL-engine view that consists of the query with the multiple tables and the use an SQL-statement referencing this view. The same method must be used, if the statement contains alternate column headings (example: "SELECT Nr AS Number.FROM...").

Decimal point

Specify the character that should be used to indicate a decimal point.

Note: The specification for an XML-Schema demands that a period must be used for the decimal point for data-types xsd:decimal, xsd:float and xsd:double. If the XML-file should be validated against an XML-Schema and a comma "," has been used for the decimal point you will receive an error message for the above data-types.

2.2.3.3.3.3 Parameters for the section "Structured text file"

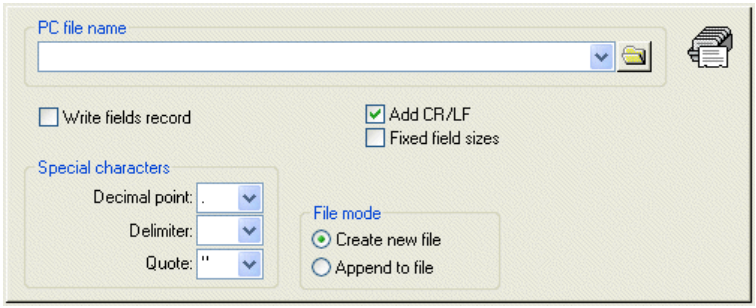


Fig. 94: Host SQL Query - Parameters for the PC data type "Structured text file"

PC file name
The specified PC file is the target for the data transfer from a host. A list of the previously used names is available after clicking the combo box button. When you click the **Search** button, a dialog box displays and allows you to browse through the directory.

Add CR/LF
Using this option adds a carriage return / line feed combination at the end of each record transferred.

Fixed field sizes
If this option is active, for each data field the defined length is applied completely (table style output). Otherwise data is saved with a minimum of resources, i.e. following blanks (character strings) or zeros at the beginning (numbers) are suppressed.

Decimal point
This field allows you to define the character that represents the decimal position on the PC.

Delimiter
The defined character is used to separate the individual fields from each other.

Quote
This character is used to enclose character string fields.

File mode
There are two options available. If the CREATE NEW FILE option is selected, an existing file is deleted before the transferred data are received. If the APPEND TO FILE option is selected, the received data is added to the existing file.

2.2.3.3.3.4 Parameters for the section "dBASE III file"

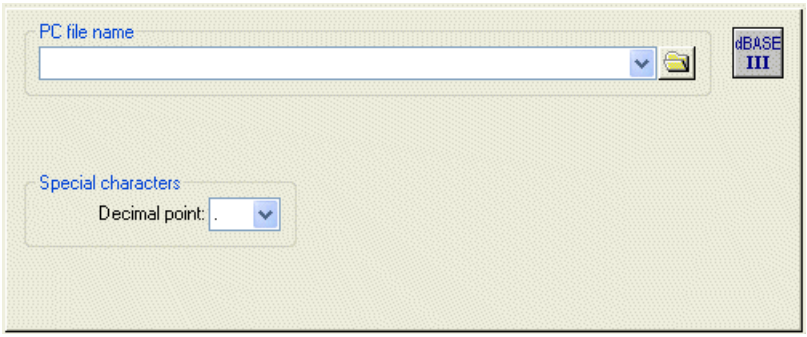


Fig. 95: Host SQL Query - Parameters for the PC data type "dBASE III file"

PC file name
The specified PC file is the target for the transfer of data from a host. A list of the previously used names is available after clicking the combo box

button. After pressing the **Search** key, a dialog is displayed and allows you to browse the directory structure.

Decimal point

This field allows you to define the character that represents the decimal position on the PC.

2.2.3.3.3.5 Parameters for the section "ODBC data"

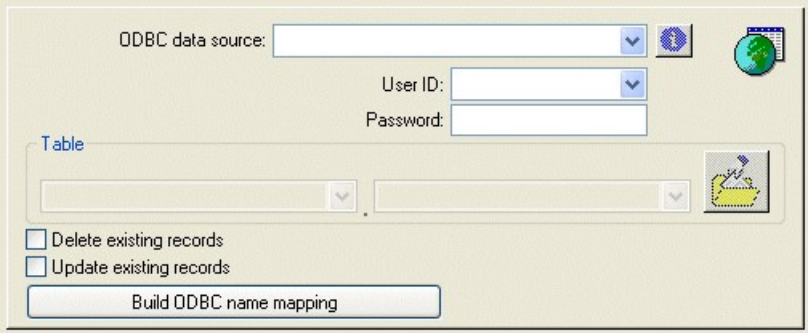


Fig. 96: Host SQL Query - Parameters for the PC data type "ODBC data"

For a description of the dialog components, see section Host => PC Transfer, page *127*.

Delete existing records

When this option is active, the records in the table will be deleted before the new records will be loaded.

Update existing records

If a **Record already exists** error occurs while inserting data into an ODBC data source and this option is active, it attempts to update the record.

Build ODBC name mapping

After a valid SQL query is entered, the fields that become active in this query is determined if this button is pressed. The result fields are inserted into the Tab Dialog "ODBC Field name mapping", page *175*). If an existing ODBC table is defined, it also searches for applicable field names. If an applicable field name is found, the key information is evaluated.

2.2.3.3.3.6 Parameter for the dataformat "Oracle"

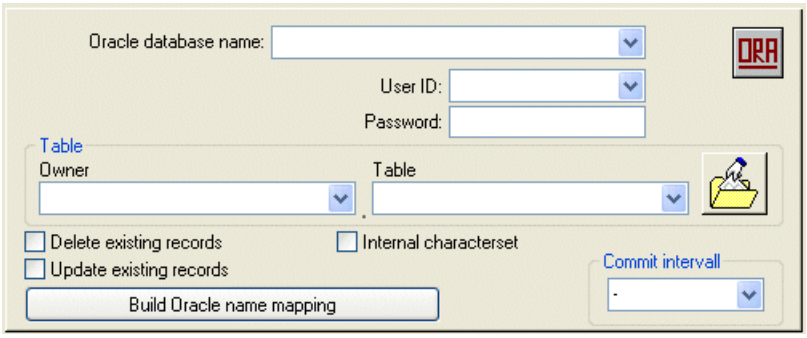


Fig. 97: Host SQL-Query – Parameter for the dataformat "ORACLE"

Oracle Database name

A "Net Service Name" must be configured on the local machine in order to access an ORACLE database using a logical database name. When an access is performed the "Net Service Name" will be resolved to the required information, i.e. network address or database instance-name. The configuration of the "Net Service Name" can be performed using the ORACLE utility "Net Manager". For further information please refer to the corresponding ORACLE documentation.

Use input field "Oracle database name" to specify the "Net Service Name" that has been configured to access the desired database. As an alternative you may also specify a connection string using the following syntax:

(DESCRIPTION=(ADDRESS=(PROTOCOL=*tcp*) (HOST=*dbhost*) (PORT=*1521*)) (CONNECT_DATA=(SERVICE_NAME=*db1*)))

Starting with databse version Oracle10g the connection string can also look like:

//host:[port][/service name]

The definition of a connection string has the advantage, that a local "Net Service Name" is not required.

User ID

Specify the name of the ORACLE user that should be used to access the database. The specification of a user id is mandatory.

Password

Specify the password for the ORACLE user.

Table

To access an ORACLE table the format of "**CREATOR.TABLE**" must be used. The owner (or creator) of the table must not be identical to the signed-on user. If no owner is specified, the signed-on user will be

considered to be the owner of the table. The input field "TABLE" however must be specified.

If the button on the right side is clicked all tables and views (and their synonyms, if available) for the specified user will be displayed. A table entry can either be selected from the list or can be manually entered in the input field.

Delete existing records

With this option active, all currently existing rows in the target table will be deleted. After that, the transfer from the host takes place.

Update existing records

If the target table already contains a record with the same key, this option ensures that the existing record will be updated. If this option has not been set and the record with an identical key already exists in the target database the insert will not take place and the source-record is being skipped.

Internal characterset

During a data-transfer ORACLE always requires the correct characterset at the local client to perform conversions from or to the database characterset.. Normally, the characterset will be obtained from the environment variables "NLS_LANG" and "NLS_NCHAR" or from the corresponding Windows-Registry entries.

For special situations option "Internal characterset" can be specified for a datatranser (HOST => PC, HOST SQL-Query, PC => HOST) and the values of "NLS_LANG" and "NLS_NCHAR" will be ignored. If this option is active, the code pages defined with the connection configuration will be used.

IMPPORTANT:

(1) If a user defined code page or code page "1160 Thai" is part of the connection configuration, the internal characterset will not be used but instead the values of "NLS_LANG" or "NLS_NCHAR" take effect.

(2) If the tcACCESS Global Language Pack is used, the complete data exchange with Oracle takes place in Unicode. In this case the settings for option "Internal character set" or for "NLS_LANG" and "NLS_NCHAR" are ignored.

The following assignemt will be used:

Code Page	Oracle-characterset
859 – Turkey	TR8MSWIN1254
870 – Easter Europe (Latin-2)	EE8MSWIN1250
875 – Greece	EL8MSWIN1253
420 – Arabic	AR8MSWIN1256
424 – Hebrew	IW8ISO8859P8

1160 – Thai	Values according to "NLS_LANG" or "NLS_NCHAR"
User defined code page	Values according to "NLS_LANG" or "NLS_NCHAR"
All other code pages	WE8MSWIN1252

Commit Intervall

Parameter "Commit Intervall" defines the number of inserted or updated rows after which a COMMIT command will be executed.. The purpose of this parameter is to increase the performance of the datatransfer. The default is that a COMMIT is executed at the end of the transfer. (after all rows have been inserted into the target-table). The default increases the processing speed for relatively small data amounts and allows to completely "ROLLBACK" in case of an error situation.. When transferring larger amounts of data it is recommended to specify a "COMMIT INTERVAL" (i.e. 100), because otherwise you may recognize a declining transfer speed during the transfer..

If "RECOVERY" has been specified, the COMMIT INTERVAL has no meaning and will be grayed out.

Build ORACLE name mapping

After a valid SQL-query has been defined clicking on this button will produce a field list of the fields that are part of the query. You can use the Tab Dialog "ODBC Field name mapping" (page *175*) to process the fields if needed. If at this point an ORACLE table has been defined that already exists, it will be searched for matching field-names. If found key-information will be evaluated..

2.2.3.3.4 Tab Dialog "ODBC Field name mapping"

If results of an SQL-query should be stored into an ODBC-or Oracle-table on the PC, the field names must be mapped. A valid SQL-query must have been defined and button "Build ODBC name mapping" or "Build ORACLE name mapping" must have been pressed (refer to Parameters for the section "ODBC data", page *172*). The tab displays all fieldnames that can now be processed by the user.

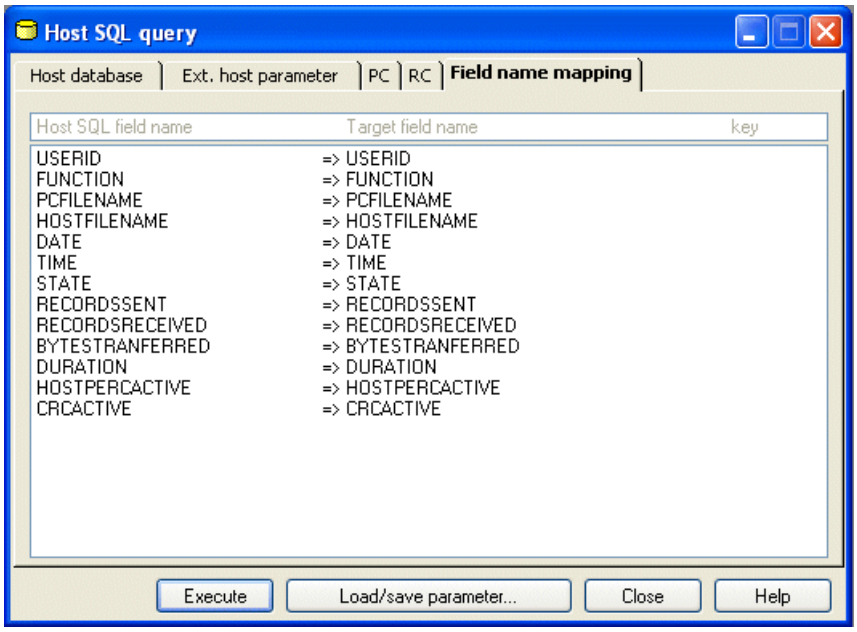


Fig. 98: Host SQL Query - Tab Dialog "Field name mapping"

The dialog components are described below:

ODBC Field name mapping list

This list contains the conversions of all host SQL field names with their ODBC- or ORACLE field name. Double-click an entry to edit it (see Edit field mapping, page *176*).

Edit field mapping

This dialog allows you to enter an ODBC- or ORACLE-field name that is the target for the evaluated host SQL field.

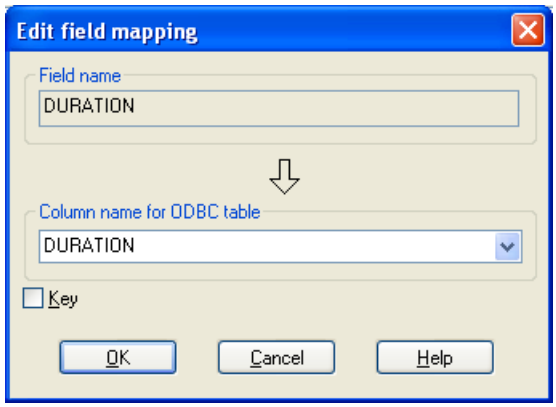


Fig. 99: Edit field mapping

The dialog components are described below:

Field name
This field shows the name of a field used on the host.

Column name for ODBC table
This field allows you to define the target ODBC– or ORACLE-column name that is used for the field above.

Key
Check this box if the ODBC- or ORACLE- field is a table key.

2.2.3.3.5 Tab "RC"

During the data-transfer, situations may emerge that will cancel the transfer (i.e. power outage, network problems etc.). This dialog can be used to restart the data-transfer exactly at that record that has been processed before the abnormal termination. This is of benefit if a larger number of records must be transferred.

The restart of an abnormally terminated data-transfer will be described as follows:

Activate the "Recovery-function" by using the check-box "Enable recovery". Only when this option is active a data-transfer can be restarted. Activating this option must take place BEFORE the actual transfer. During the transfer a recovery file is being created which contains status information of the data being transferred.

After each data-transfer you can find information about the transfer ("Status: Transfer successfully completed" or "Status: Transfer uncomplete"). If the transfer has been successful, the recovery file will be overwritten during the next transfer. "Transfer type" will be "Normal".

In case of an unsuccessful transfer you can load the recovery file (file-type "rsq") like a normal parameter file using the button "Load/Save parameter" and the command "Load parameter...". You must select the file type "Recovery Parameter (*.rsq)" to display the "rsq"-file. After loading the file the original file-transfer parameter will be loaded and additional status information will be processed. "Transfer type" is "Recovery" indicating that when the file transfer will be startet an actual restart will take place. If the new transfer also fails, you can again load the "Recovery-file" and restart the transfer. The transfer can be repeated as often as necessary in order to complete the transfer.

Note: After a "Recovery-file" has been loaded, the dialog parts "Enable recovery " or "Recovery-file name" are greyed out and cannot be modified. "Transfer type" is "Recovery". If you decide to not restart the transfer. If you do not want to perform the restart function you can use the "Reset"-button to change the transfer type to "Normal". This will enable new transfers with new settings. The status information will also be reset.

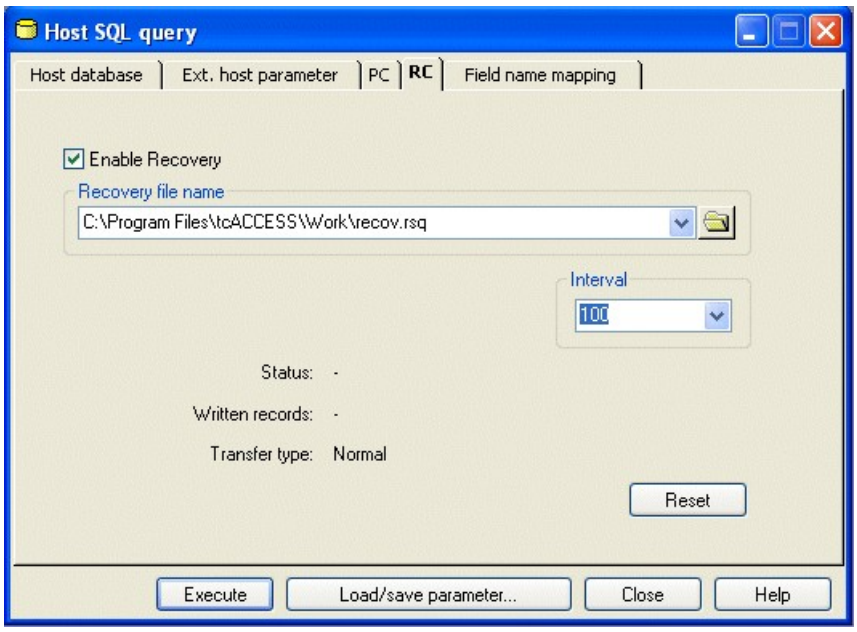


Fig. 100: Host SQL-Query - Tab "RC"

The dialog components have the following meaning:

Enable recovery

With this option acitve, status information will be written into a "Recovery file" that allows the restart of a data-transfer. Path and file name of the recovery-file (file type "rsq") must be specified.

Recovery file name

Status information about the data transfer will be saved to that file. You can select the file name from the list of previously used file names. You can also browse for the file name by clicking on the corresponding symbol.

To be able to use a recovery file, option "Enable recovery" must be activated before the data-transfer.

Interval

The value specifies, how often a recovery entry is written to the recovery file according to the number of written records of the data transfer. Increasing the value for "Interval" decreases the influence of the recovery

on the performance of the data transfer, but if a high value is specified, possibly many records have to be rewritten in case of a recovery transfer. The default value for "Interval" is 100.

Note: The value of "Interval" has no effect for the PC data type "ODBC data".

Status

If option "Enable recovery" is active, information will be displayed indicating whther the data-transfer has been successful or not.

Written records

Specifies the number of records written to the target file or target-table.

Transfer type

"Recovery" will be displayed, when the recofery file has been loaded after an abnormally completed data-transfer. "Normal" will be displayed in all other cases or when the "Reset"-button has been used.

Reset

If a recovery fiule has been loaded to restart a transfer, dialog parts "Enable recovery" and "Recovery file name" are greyed out and cannot be changed. Clicking on the "Execute"-button of the dialog "Host -> PC-Transfer" the data-transfer will be restarted using the settings and status information of the recovery file. If you do not want to restart the transfer you can use the "Reset"-button. The transfer type will be reset to "Normal" and "Enable recovery" and "Recovery file name" can be modified. All status information displayed under tab "RC"will be reset.

Note: The recovery function is not available for the "*Preview*"-format. If a query will be executed using this format and the "Enable recovery" check box has been activated, it will be automatically deactivated. For a subsequent query to a non "Preview"-format the option must be set again, if recovery should be used.

2.2.3.4 Virtual Disks

You can open this dialog by selecting the appropriate menu item or clicking the button shown below. It allows you to connect virtual disks with PC disk drive letters, and format and delete them. Virtual disks can be accessed either via the tcACCESS Command Shell (*see The tcACCESS Command Shell BMLSHELL*, page *315)* or the functions PC => Host Transfer (see page *97*) or Host => PC Transfer (see page *127*).

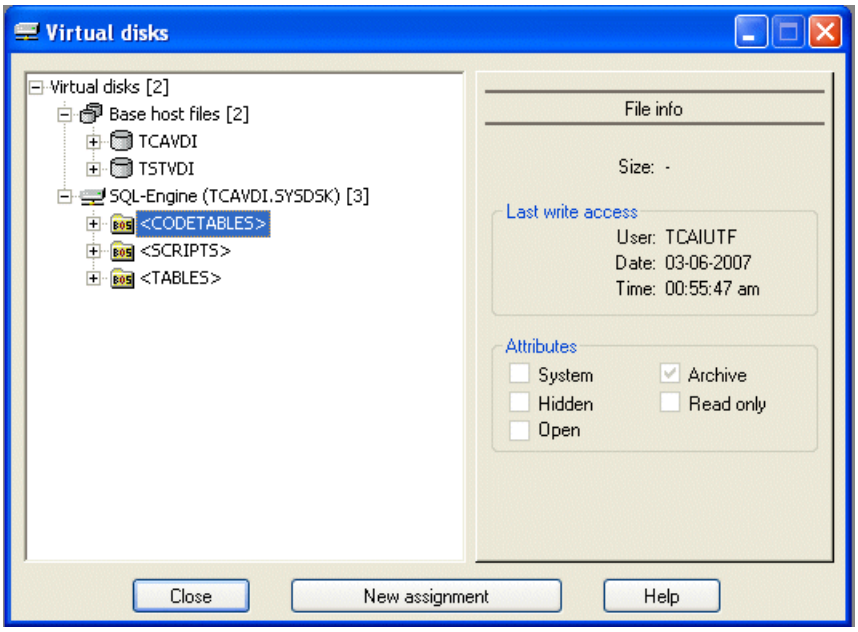


Fig. 101: Administration of Virtual Disks

The dialog components are described below:

Tree view

The tcACCESS host version 3.01 virtual disk of the SQL-Engine displays in the tree view and shows which PC disk drive letters it is connected to. With version 4.0, the tcACCESS host components or Base host files also display. The tree view shows all base files for the virtual disks (of version >= 4.0) which were defined during the host generation.

If an entry is opened, tcACCESS retrieves the related files and directories of the base directory from the mainframe.

When you select a base host file, which contains the partitions that correspond to the virtual disks, then right mouse click, the following entries display:

Update

Retrieves the partitions of the base host files again.

Format virtual disk

This function allows you to create a new virtual disk (partition) within the selected base host file.

When you right-click on a virtual disk entry, the following context menu opens:

Update

Retrieves the files of the virtual disk files again.

Copy PC file into directory
This function allows to copy a PC file into the base directory of the virtual disk

Create directory
Allows the creation of new directories on the Virtual Disk

Format virtual disk V3
This menu item is only available, if a host connection to a tcACCESS version less than V5 exists. See, page *182*

Purge virtual disk
After confirmation, the selected partition is removed from the base host file.

Cancel assignment
The disk drive letter connected to the selected virtual disk is released. This has no effect on the host.

Change host file
See The Dialog "Change host file", page *183*

Right click on a directory entry within a virtual disk and the following context menu opens:

Update
Retrieves the virtual disk directory, where the file is located, again.

Copy PC file into directory
This function allows to copy a PC file into the base directory of the virtual disk

Create directory
Allows the creation of new directories on the Virtual Disk

Delete
This function allows you to delete a file on a virtual disk.

Right click on a file entry within a virtual disk and the following context menu opens:

Update
Retrieves the virtual disk directory, where the file is located, again.

View
Copies the file to the PC and displays the contents using the "Program for Text Display" option setting. (see Options page 93).

Copy to PC
Copies the selected file to a PC.

Rename
Renames the selected file

Delete
Deletes the selected file

File infos
If a directory or a file of a virtual disk is selected in the tree view, this group displays the related file information. This includes the size of the file in bytes, information about the last write access, and the file attributes.

Virtual disk infos
If a virtual disk is selected within the tree view, related information is displayed there. This includes the current status, total and free disk space, and information about the creation of the partition.

Close
After clicking this button, the dialog ends.

New assignment
Pressing this button opens a dialog to connect a virtual disk to a PC disk drive letter. A detailed description can be found in section The Dialog "New assignment", page *184*).

Help
Displays online help for the open dialog.

2.2.3.4.1

2.2.3.4.2 The Dialog "Format virtual disk"

This dialog allows you to create a virtual disk (partition) within an existing base host file.

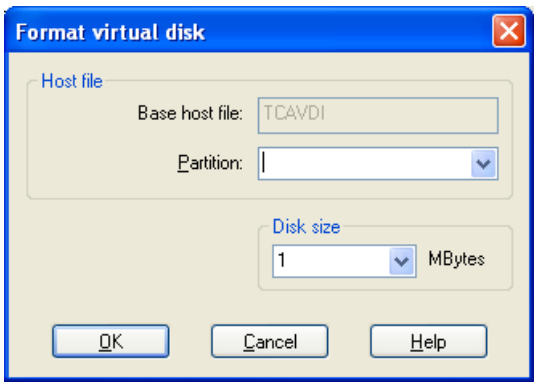


Fig. 102: Virtual Disks - Dialog "Format Virtual Disk"

The dialog components are described below:

Partition
Name of the partition (virtual disk) that is created.

Disk size
Size of the new virtual disk in megabytes.

OK
The virtual disk shown in the upper part of the dialog is formatted based on the size specified in the DISK SIZE field.

Cancel
Terminates the dialog without performing an operation.

Help
Displays online help for the open dialog.

2.2.3.4.3 The Dialog "Change host file"

This dialog allows you to connect an assigned disk drive letter to another host file. Depending on whether the mainframe operating system is VM or not, the file can be defined BY FILE NAME, FILE TYPE and FILE MODE (CMS file) or by the BASE HOST FILE and PARTITION (VSAM files).

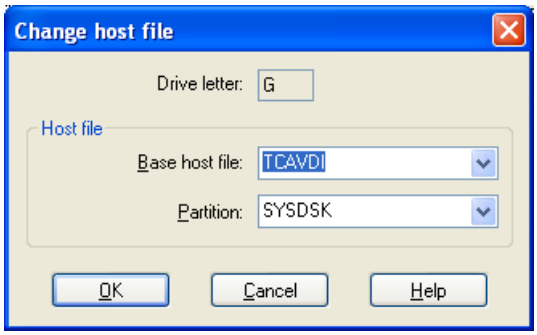


Fig. 103: Virtual Disks - Dialog "Change Host File"

The dialog components are described below:

Base host file
This input field is available for any IBM host operating system except VM. A VSAM file is internally separated into individual partitions. Each partition is a virtual disk. Its name can be up to 8 characters long.

Partition
This input field is available for any IBM host operating system except VM and refers to the part of the BASE HOST FILE that is identical to a virtual disk. Its name can be up to 8 characters long.

File name, file type, file mode

These input fields are only available with the host operating system VM. They can be used to specify a CMS file that contains the virtual disk.

OK

After clicking this button, the defined disk drive letter is assigned to the specified virtual disk.

Cancel

Terminates the dialog without performing an operation.

Help

Displays online help for the open dialog.

2.2.3.4.4 The Dialog "New assignment"

This dialog opens after clicking the appropriate button within the main dialog of the virtual disks. It allows you to assign a virtual disk (host file) to a PC disk drive letter, which can then be addressed from the *tcACCESS Command Shell* (see The tcACCESS Command Shell BMLSHELL, page *315*) or by the PC => Host Transfer, page 97) or Host => PC Transfer, page *127*) functions. If the DISK SIZE input field is filled, the virtual disk is also formatted.

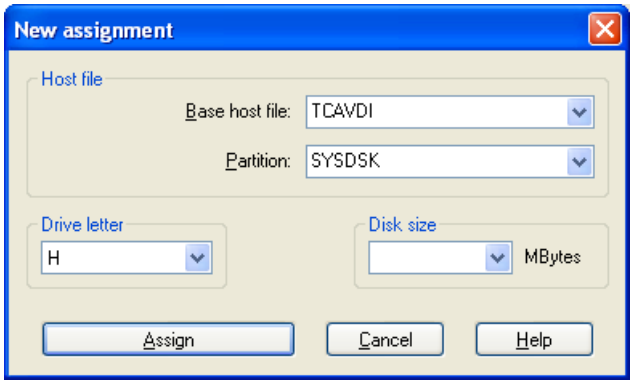


Fig. 104: Virtual Disks - Dialog "New Assignment"

The dialog components are described below:

Base host file

This input field is available for any IBM host operating system except VM. A VSAM file is internally separated into individual partitions. Each partition is equivalent to a virtual disk. Its name can be up to 8 characters long.

Partition

This input field is available for any IBM host operating system except VM, and referes to the part of the BASE HOST FILE that is identical to a virtual disk. Its name can be up to 8 characters long.

File name, file type, file mode

These input fields are only available with the host operating system VM. They can be used to specify a CMS file that contains the virtual disk.

Drive letter

The combo box displays all disk drive letters on the local PC which do not have a physical or logical drive assigned. A virtual disk can be addressed from *the tcACCESS Command Shell* (see The tcACCESS Command Shell BMLSHELL, page *315*) via this letter.

Disk size

If this field contains a value, the **Assign** button changes to **Assign and Format**. The disk size specified in megabyte is used to format the virtual disk. If the field is empty, the virtual disk is not formatted before the assignment.

Assign / Assign and Format

If this button is selected, the base host file and partition specified is assigned to a PC disk drive letter. If the DISK SIZE field is also specified, the virtual disk is formatted according to the defined size.

Cancel

Terminates the dialog without performing any operation.

Help

Displays online help for the open dialog.

2.2.4 The Menu "Administrator"

When you start tcACCESS, the only Administrator menu entry available is the ADMINISTRATOR LOGON. Once the administrator signs on successfully, all other menu items are available. All settings modified in these menu items (except tcACCESS Host Security Administration, page *206*) are stored in the tcACCESS.cnf file in encoded form.

Unless otherwise noted, the buttons on each dialog in this section have the following functions:

MENU ITEMS	REFERENCE
OK	Values entered in a dialog are verified and saved in the configuration file.
Cancel	Closes the dialog without saving any modified values.
Help	Displays online help for the open dialog.

2.2.4.1 Administrator Logon

2.2.4.1.1 Tab Dialog "Password"

Use this tab dialog to enter the valid administrator password for the tcACCESS client system on the PC.

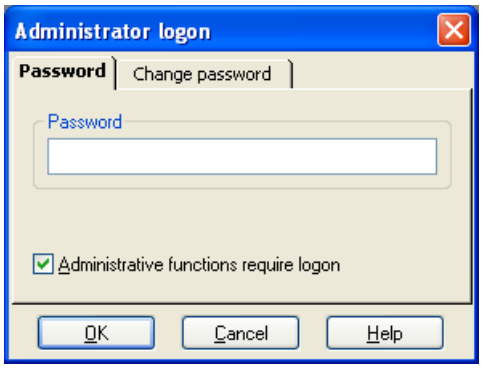


Fig. 105: Administrator Logon - Tab Dialog "Password"

The dialog components are described below:

Password

Use this field to enter the administrator password as defined during tcACCESS installation. After installation, the standard password is **admin**. However, it is recommended that you change this standard identification into an individual entry. Use the CHANGE PASSWORD tab dialog for this purpose (see below).

Entries into this field appear as asterisks ("*") for security reasons.

Administrative functions require logon

This option is set after installing tcACCESS. It requires a logon with the administrator password before you can perform any operations within the sub-menu. If you do not want this security restriction, use this button to skip it. In this case all menu items are immediately available after tcACCESS starts.

2.2.4.1.2 Tab Dialog "Change password"

This dialog allows you to change the administrator password for the local PC if the PASSWORD tab dialog contains a valid entry.

Fig. 106: Administrator Logon - Tab Dialog "Change password" - Status Invalid

Fig. 107: Administrator Logon - Tab Dialog "Change password" - Status OK

The dialog components are described below:

New password

This field allows you to change the administrator password. To process this function it is mandatory that the administrator has signed on and the desired password is entered correctly a second time in the VERIFY NEW PASSWORD field.

Verify new password

The **OK** button is only available when the character string in this field is identical to input in the NEW PASSWORD field. If this is not the case, the STATUS: INVALID message displays. If the input is correct, the message changes to STATUS: OK and the **OK** button becomes available.

2.2.4.2 Settings

The SETTINGS dialog is only available when the administrator has signed on correctly (see Administrator Logon, page *186*). It allows you to customize various system-related options, which are not accessible for every user. The defined values are stored in the tcACCESS.cnf file in encoded form.

2.2.4.2.1 Tab Dialog "General"

This tab dialog provides general settings for use by the administrator.

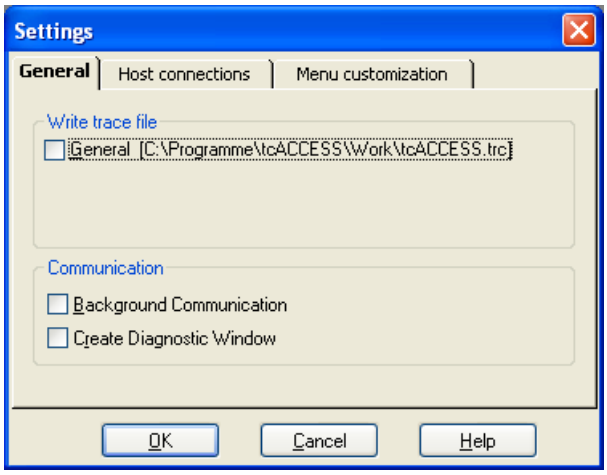


Fig. 108: Settings for Administrators - Tab Dialog "General"

The dialog components are described below:

Write trace file - General

After you check this option, general trace information is written into the file, which displays behind it. If a problem occurs, this file may be sent to the tcACCESS support team.

Background Communication

Using this option, the communikation handler will not be displayed. This compares to "running as service".

Create Diagnostic Window

This option checked means, that the diagnostic window of the communikation handler is being created and filled for diagnostic purposes.

2.2.4.2.2 Tab Dialog "Host connections"

Use this tab dialog to customize the allocation of a host session and what connection slots should be used by a user.

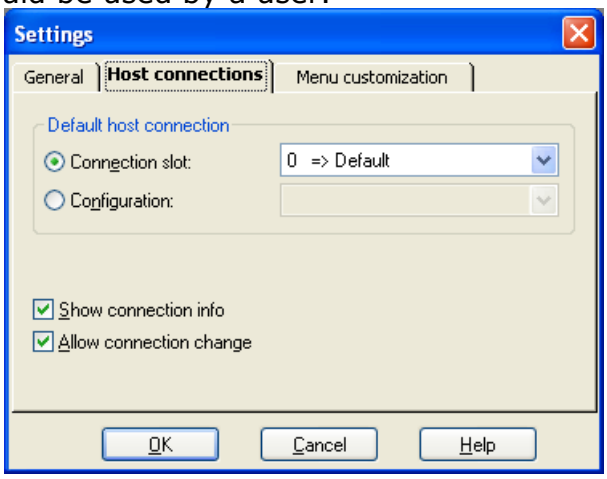


Fig. 109: Settings for Administrators - Tab Dialog "Host connections"

The dialog components are described below:

Default Host Connection

You can specify the connection which should be used as default. This slot will also be used by the ODBC driver, if the driver option "Standard Host connection" has been selected.

If you select option "Connection slot" the specified slot will be fixed as the standard slot. The user cannot chose another slot. However, if you select option "Configuration", all slots can be used which are assigned to the specified configuration.

A user is limited to these settings only, if the check box "Allow connection change" is not active.

Show connection info

If this option is checked, the logon dialog displays information about the desired connection. Selection of a connection ist not allowed by this option.

Allow connection change

If option "Show connection info" is checked, this option can be checked additionally. It enables the user to change the connection slot.

2.2.4.2.3 Tab Dialog "Menu customization"

This tab dialog contains all available menu items. These items can be individually restricted for the end user. They can be enabled again by accessing the administrator functions.

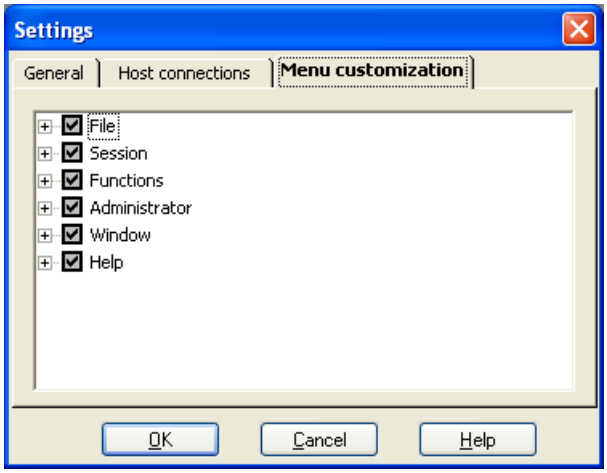


Fig. 110: Settings for Administrators - Tab Dialog "Menu customization"

You cannot disable the "Exit" menu item and all "Administrator" Menu items. These items can only be accessed, if you are signed on as an administrator. Menu items related to the configured connection slots cannot be disabled either.

The specified values are saved to "tcaccess.cfg" in a compressed format.

The dialog components are described below:

Tree view

This dialog shows the menu structure. The little mark on the left side of the menu item indicates, that the menu item is enabled. If the color of the box is grey indicates, that no changes can be applied to this menu item.

OK

The settings will be accepted and saved.

Cancel

The changes to the settings will not be saved.

Help

Display of the online help dialog.

2.2.4.3 Connection configurations

This dialog enables an user who is logged on as administrator to change the configuration of the host connection (see Administrator Logon, page 186). Different settings can be saved and assigned to slots.

Concept:

The administration of the connection has a two step concept. First, all necessary information is saved as a configuration with an individual name. The configuration includes the following settings:

- type of connection
- logon/logoff sequence for a terminal-connection
- system tables for data access
- code pages

The configurations can be created, edited, renamed and deleted with the dialog as explained in this chapter. The configurations are saved with the user assigned names plus the extension ".slc" in the config subdirectory of the tcACCESS installation.

The second step of the connection administration is to assign these configurations to slots. Now you can choose a slot to be used for the connection and the data transfer. Up to 255 configurations can be assigned to slots. The product license only affects and limits the number of simultaneously established host connections.

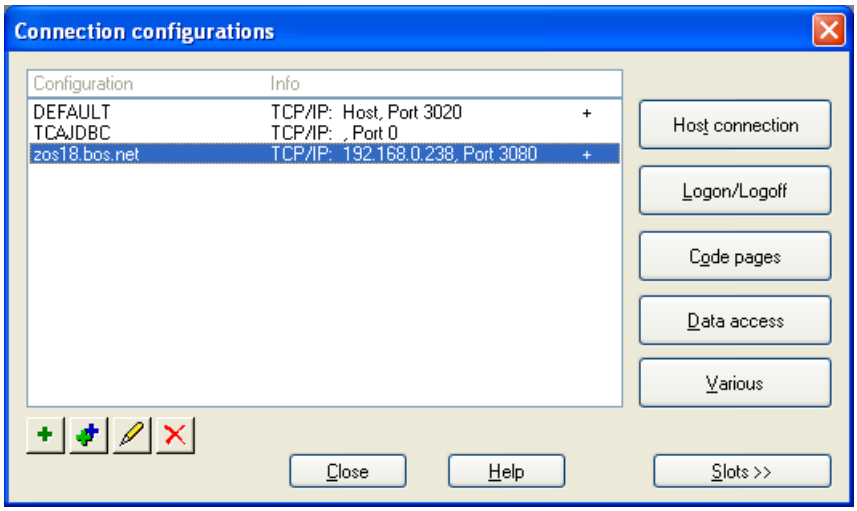


Fig. 111: Settings of Connection configurations

The dialog components are described below:

List of connection configurations

The list shows all configuration files (extension ".slc") of the subdirectory "config" in alphabetic order. A short version of the settings is shown for every configuration. If a connection is already assigned to at least one slot, a "+" is displayed at the right side of the line. Several buttons are provided for editing the settings.

The iconic buttons are described below:

 Create a new configuration

 Copy the selected configuration

 Rename the selected configuration

 Delete the selected configuration

Host Connections

This button shows a dialog for specifying the wanted type of host connection (see Host Connections, page 192).

Logon/Logoff

Configuration of the user logon/logoff (see Logon/Logoff, page 193)

Code Pages

In this dialog you can specify the ASCII/EBCDIC translation settings. (see ASCII/EBCDIC Code Pages, page 201).

Data Access

This dialog contains all the settings related to structured downloads and SQL. (see Data Access, page 203)

Various

Here you can specify the timeout and data transmission settings. (see Various, page 204).

Close

Closes the dialog.

Help

Shows the online help according to the displayed dialog.

Slots >>

Switches to the Connection Slots dialog (seeConnection Slots, page 205)

2.2.4.3.1 Host Connections

This dialog allows you to determine the type of connection to the mainframe. Depending on the connection type, the available parameters are displayed in the right section of the dialog.

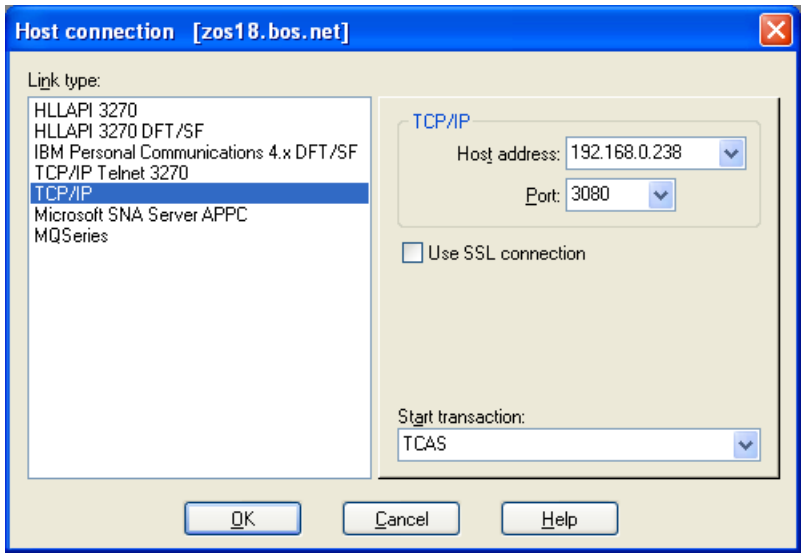


Fig. 112: Host Connection Setup –

The dialog components are described below:

Link type

Use this option to define the desired type of connection to the host. An overview on the supported methods for the individual operating systems can be found under section Supported Types of Host Connections, page *1*.

Link type related parameters

The right section of the tab dialog displays the various setting options for a selected link type.

2.2.4.3.2 Logon/Logoff

All settings within this dialog are related to user logon/logoff to the mainframe.

2.2.4.3.2.1 Tab Dialog "General"

This tab dialog deals with general settings related to user sign on to the mainframe.

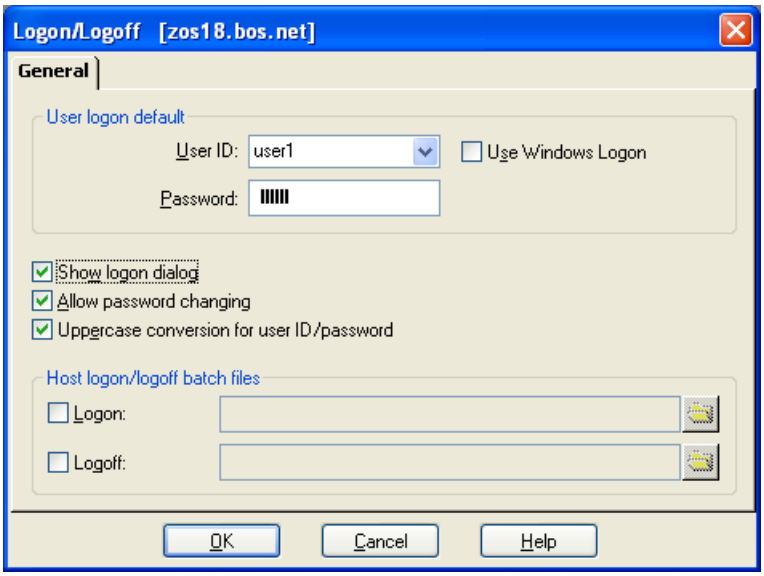


Fig. 113: Tab Dialog "Logo/Logoff sequence/General"

The dialog components are described below:

User logon default

You can specify a default user id and password.

Use Windows Logon

If this option is active, the Windows sign on name will always be used for the host logon. "NOPASSWD" will be used as password.

Show logon dialog

If you do not check this option, no user logon dialog is open when a session is allocating. This can be useful for host connections without authentification (e.g., via 3270 host emulations). Otherwise, you may use this feature by providing the necessary logon info in the appropriate input fields of the USER LOGON DEFAULT group.

Allow password changing

It is necessary that this option is active to change the user password on the host during the start of a session. Otherwise the related tab dialog is deactivated. If a command line parameter has been applied, it is ignored.

Uppercase conversion for user ID/password

You should activate this option if the connected mainframe accepts the input for user and password in capital letters only.

Host Logon/Logoff batch files

If you want to use your own batch files instead of the logon/logoff sequences, you can specify< them here. If you use batch files, the tabs for logon and logoff sequences will be deactivated.

2.2.4.3.2.2 Tab dialog "Logon/Logoff sequence”

This tab dialog is only available after selecting a PC Communication Handler that requires a 3270 host emulation (i.e., not with TCP/IP Winsocket and MS SNA Server APPC).

The LOGON SEQUENCE allows you to automatically process the 3270 commands for a host logon. If the tcalogon.bat file (see also example file tcalogon.sam) exists in the tcACCESS installation directory, it is processed instead of the logon sequence defined here.

A "+” sign at the tab indicates, that a corresponding sequence has been defined.

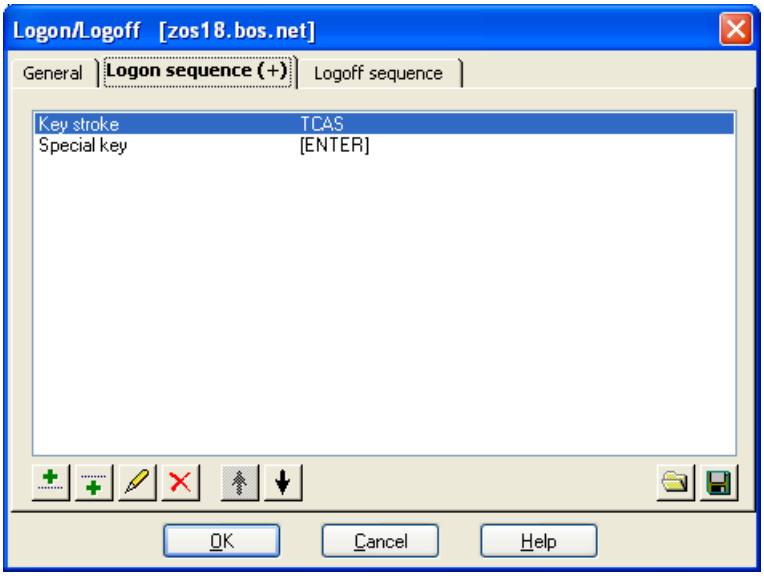


Fig. 114: Tab Dialog "Logo/Logoff sequence"

List of 3270 commands to be processed

The list contains the 3270 entries that are used to establish a connection via a host emulation. A double-click on an entry corresponds to clicking the **Edit** button.

The last two commands should always be a **TCAS** key input followed by the **ENTER** key in order to start the tcACCESS Host Communication Handler required for the connection.

The icons are described below

 Add a new entry in front of the selected entry

 Append a new entry after the selected entry

 Edit the selected entry

 Delete the selected entry

 Move the selected entry one position up


Move the selected entry one position down


Load a logon/logoff sequence from a sequential file. This can also be a "tcACCESS.cnf" file (tcACCESS configuration file up and until Version 4).


Save the logon/logoff sequence to a sequential file. This file may be processed using any conventional text editor and may be used for individual Logon- or Logoff-Batch files.

2.2.4.3.2.3 "Edit 3270 command" Dialog

This dialog contains all supported types of 3270 commands and some special input data provided by the user's host logon input.

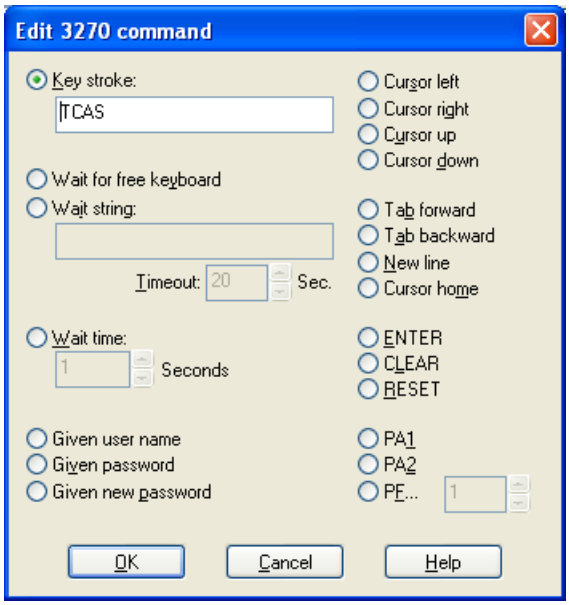


Fig. 115: Logon/Logoff Sequence - Dialog "Edit 3270 command"

The dialog components are described below:

Key stroke

All input in this field is interpreted as *normal* keys. The input of small and capital letters is considered.

Wait string

Processing a 3270 command sequence is stopped due to an input in this field either until the defined character string appears in the 3270 screen buffer or until the timeout interval defined under Settings (see page *187)* is exceeded.

Wait for free keyboard

Use this option for waiting until the keyboard is not blocked anymore. This option is useful, if the wait time depends on the connection or the time of the day. It is only available with TN3270 Telnet connection type.

Wait time

If there is no fixed character string, which is expected to display, you can define a wait time. It can be a value between 1 and 99 seconds.

User input

In the following three fields—GIVEN USER NAME, GIVEN PASSWORD, and GIVEN NEW PASSWORD— insert the appropriate character string at the current cursor position of the logical 3270 buffer string. For this, the values are used that have been defined at the start of the session (see Allocate Session, page *89*).

Special key

All other options correspond to a special key on the 3270 terminal keyboard. The following keys are supported:

- Cursor left
- Cursor right
- Cursor up
- Cursor down
- Tab forward
- Tab backward
- New line
- Cursor home
- ENTER
- CLEAR
- RESET
- PA1
- PA2
- PFxx

You can specify the desired function key by entering a number into the input field next to it, for example, PF1 - PF24.

OK

The defined 3270 command is returned to the calling window and the dialog ends.

2.2.4.3.3 Syntax description of the 3270 Interface

The commands of the 3270 interface have the following format:

[CPPP...]

Where 'C' stands for one of the following commands:

K	Simulate key input
C	Copy character string to screen buffer

A Simulate an AID key

S Wait for screen output

R Read 3270 screen buffer

E End of command sequence

and 'PPP' represents a parameter.

Simulate key input
The ASCII characters following 'K' are sent like key inputs to the control unit which will customize the 3270 screen buffer then.
By adding a '@' character to the key list, input keys can be simulated which do not correspond with an ASCII character. The following combinations are valid:

INPUT KEY	ACTION
@L	Cursor left
@R	Cursor right
@U	Cursor up
@D	Cursor down
@F	Tabulator forwards
@B	Tabulator backwards
@E	Delete end of line
@N	New line
@H	Cursor to base position
@I	Reset
@Wxx	Wait; thereby xx defines the number of seconds to be waited (e.g., @W04)
@@	The "@" character

Example:
[K@H@Ifield1@N@Iline2@H@B@Ilast field@@]

Submit this character sequence to the 3270 interface, to obtain the following operation results:
1. Cursor to base position
2. Reset
3. Key input "field1"
3. New line
5. Reset
6. Key input "line2"
7. Cursor to base position
8. Tabulator backwards
9. Reset

10. Key input "last field@"

Note:

It is recommended that you use a reset command before the beginning of a key input sequence to avoid any keyboard lock problems.

Copy character string to screen buffer

The ASCII characters following "C" are copied to the screen buffer at the current cursor position.

Simulate an AID key

The two characters following "A" determine the AID key to be simulated. The following combinations are valid:

AID KEY	DESCRIPTION
EN	<ENTER>
CF	<CLEAR>
01	<PF1>
02	<PF2>
....	
24	<PF24>
A1	<PA1>
A2	<PA2>

Example:

[**AEN**]

The character string above simulates clicking the **ENTER** key on a 3270 keyboard.

After sending an AID key, a "Wait for screen output" or a "Read 3270 screen buffer" command should always follow in order to wait for the response from the host.

Wait for screen output

This function allows a PC application to wait until one or more messages display in the 3270 screen buffer. The command must be structured as follows:

[S[string0^string1^...^string19]

The "^" character separates the individual character strings which should be waited for. It is possible to enter a number between 1 and 20 on these character strings.

When none of these character strings appear in the screen buffer during the timeout time, a host timeout error is returned in the Communication Handler return code to the PC.

If one of the character strings is found in the 3270 screen buffer, the reply parameter buffer contains a 4-bytes long ASCII number which identifies the character string that is found. The counterpart for the first character string is number "0".

Character strings are searched for according to the defined sequence. The first character string that matches, ends the search process. If the last defined search string is empty, the search process is immediately ended at this position, when none of the preceding character string has matched. This method may be

applied to test, if the screen buffer currently contains one of the previously defined character strings.

Example:

[S[MSG1^MSG]

The value specified in the request parameter buffer waits for the occurence of either the character string "MSG1" or "MSG" within the 3270 screen buffer. When "MSG1" is found, the character sequence "0000" is returned; if message "MSG3" occurs, the character sequence "0001" is returned.

Read 3270 screen buffer

The content of the 3270 screen buffer is copied to the reply data buffer; the reply parameter buffer contains the current cursor position in the form of a 4-bytes ASCII number. Thereby, the counterpart to the left upper screen corner is "0".

Example:

[R]

The above input into the request parameter buffer causes the Communication Handler to copy the 3270 screen buffer into the reply data buffer and to write the current cursor position to the reply parameter buffer. This function waits until the system clock sign in the 3270 status bar disappears; this, however, does not match with the desired screen content under certain circumstances:

- Certain 3270 applications (e.g., VM/CMS) delete the system clock sign before they show the complete host response.
- In combination with certain asynchronous connections no status bar exists. Then the Communication Handler can not differentiate whether the host response is complete or incomplete. In this case the Communication Handler assumes that the screen setup is complete, when the first byte of the host response has been received and two or more seconds are passing without receiving additional data in the meantime.

In these cases, it is recommended that you use the WAIT FOR SCREEN OUTPUT function which is described earlier in this chapter.

End of command sequence

A sequence of the commands described in the previous sections can be passed in a request parameter buffer. This command ends the sequence. It can be defined optionally at the end of the request parameter buffer as the end of processing command in all cases.

Example:

[E]

Ends the command sequence

Note:

The commands described in the previous sections can be added in a request parameter buffer. See the following example:

[ACF] [K@ICSSN] [AEN] [S[NAME] [R] [E]

This command sequence simulates pressing the **CLEAR** key, enters the character string CSSN at the current cursor position, simulates the **ENTER** key, waits for the screen output NAME, and finally reads the 3270 screen content into the reply data buffer.

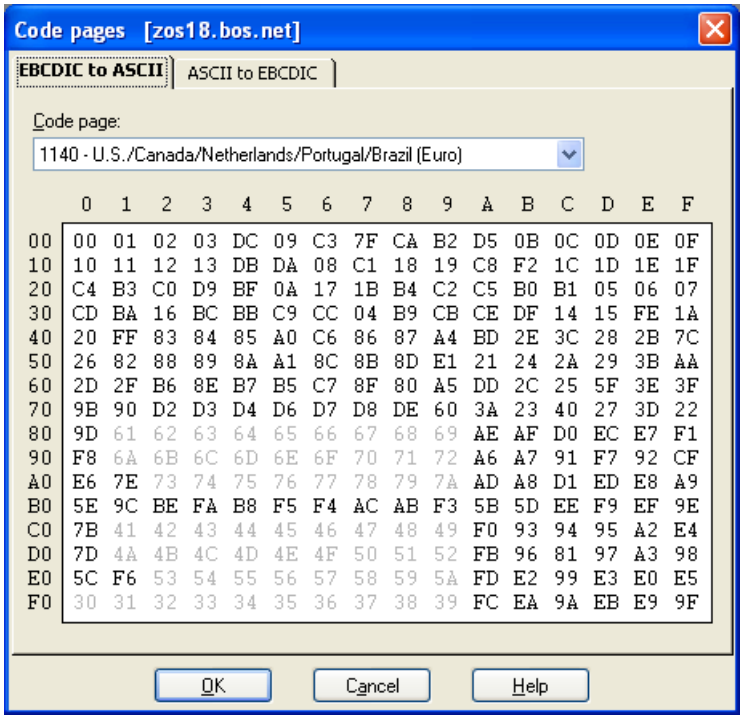
Note:

If more than one 3270 command on the command sequence writes information into one of the reply buffers, the old values are overwritten (e.g., a "Read 3270 screen buffer" command overwrites the search character string number of a preceding "Wait for screen output" command with the return of the current cursor position).

2.2.4.3.4 ASCII/EBCDIC Code Pages

This dialog is only available after a correct administrator signon (see Administrator Logon, page *186*). It allows you to define the desired code pages for the exchange of data between host (EBCDIC) and PC (ASCII). Depending upon the configuration, the defined values are saved in the corresponding ".slc" file.

When using the "Global Language Packs" these codepages are no longer relevant for SQL acces with the Multibyte-feature. The transfer happens automatically based upon UTF-8 or UNICODE.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	01	02	03	DC	09	C3	7F	CA	B2	D5	0B	0C	0D	0E	0F
10	10	11	12	13	DB	DA	08	C1	18	19	C8	F2	1C	1D	1E	1F
20	C4	B3	C0	D9	BF	0A	17	1B	B4	C2	C5	B0	B1	05	06	07
30	CD	BA	16	BC	BB	C9	CC	04	B9	CB	CE	DF	14	15	FE	1A
40	20	FF	83	84	85	A0	C6	86	87	A4	BD	2E	3C	28	2B	7C
50	26	82	88	89	8A	A1	8C	8B	8D	E1	21	24	2A	29	3B	AA
60	2D	2F	B6	8E	B7	B5	C7	8F	80	A5	DD	2C	25	5F	3E	3F
70	9B	90	D2	D3	D4	D6	D7	D8	DE	60	3A	23	40	27	3D	22
80	9D	61	62	63	64	65	66	67	68	69	AE	AF	D0	EC	E7	F1
90	F8	6A	6B	6C	6D	6E	6F	70	71	72	A6	A7	91	F7	92	CF
A0	E6	7E	73	74	75	76	77	78	79	7A	AD	A8	D1	ED	E8	A9
B0	5E	9C	BE	FA	B8	F5	F4	AC	AB	F3	5B	5D	EE	F9	EF	9E
C0	7B	41	42	43	44	45	46	47	48	49	F0	93	94	95	A2	E4
D0	7D	4A	4B	4C	4D	4E	4F	50	51	52	FB	96	81	97	A3	98
E0	5C	F6	53	54	55	56	57	58	59	5A	FD	E2	99	E3	E0	E5
F0	30	31	32	33	34	35	36	37	38	39	FC	EA	9A	EB	E9	9F

Fig. 116: Code Pages

The dialog components are described below:

Code page

Here you can select the desired code page. It displays in the lower part of the dialog. The list contains the following tables, with a standard number at the beginning:

- International
- 037 - USA/Canada/Netherland/Portugal/Brazil
- 273 - Germany/Austria
- 277 - Denmark/Norway
- 278 - Sweden/Finland
- 280 - Italy
- 284 - Spain
- 285 - United Kingdom

- 297 - France
- 500 - International
- 1140 USA/Canada/Netherland/Portugal/Brazil (Euro)
- 1141 Germany/Austria (Euro)
- 1142 Denmark/Norway (Euro)
- 1143 Sweden/Finland (Euro)
- 1144 Italy (Euro)
- 1145 Spain (Euro)
- 1146 United Kingdom (Euro)
- 1147 France (Euro)
- 1148 International (Euro)
- User modified code page

  As soon as a table value is modified by a user, the current list selection is positioned on the appropriate entry. If there is no standard table to apply, the USER MODIFIED CODE PAGE is selected.

Hexadecimal display of a selected code page

The selected table in the CODE PAGE field is displayed in the lower part of the dialog in the form of hexadecimal values.

Clicking on an entry shown in black opens the window, which is described in *section "Change code page entry", page 202*. The window does not open if the values selected are grey, since these entries must remain constant.

2.2.4.3.4.1 "Change code page entry" Dialog

This dialog allows you to alter a value of a selected code page.

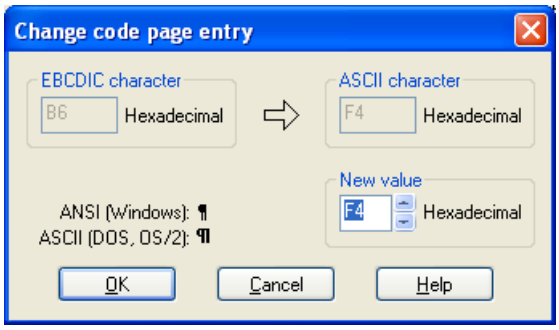


Fig. 117: Dialog "Change code page entry"

The dialog components are described below:

EBCDIC character / ASCII character

Depending on whether the EBCDIC to ASCII or the ASCII to EBCDIC table is processed, a description of the EBCDIC and ASCII characters displays.

The left field shows the value to be translated; the upper field on the right indicates the currently defined result of the translation.

New value

This field allows you to define a new translation for the character value, which is defined in the left field. It is possible to enter a value between 0 and FF Hex (equivalent to the decimal number 255).

OK

The code entered into the NEW VALUE FIELD is used in the related code page.

2.2.4.3.5 Data Access

This dialog contains all the settings related to structured downloads and SQL.

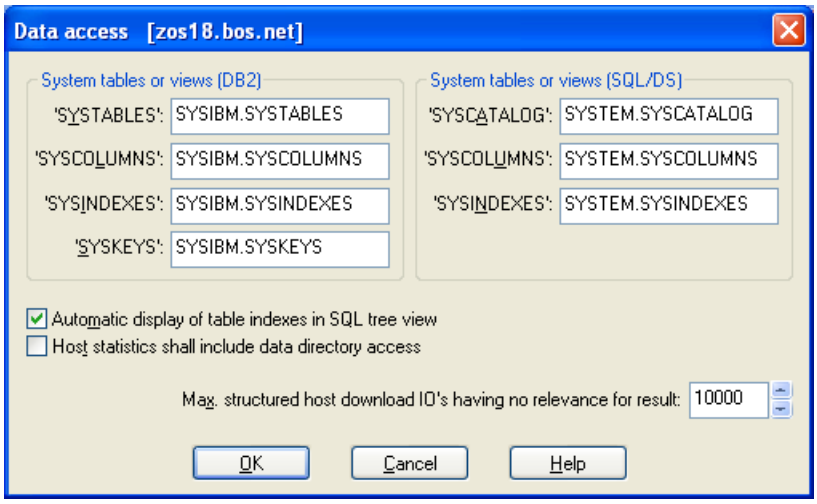


Fig. 118: Dialog "Data Access"

The dialog components are described below:

DB2 system tables (Views)

This input fields are needed for the DB2 (under OS/390) tree view within the Host SQL Query dialog (see page *156*).

Since the user who is logged on to the host may not be authorized to read the system tables these fields are helpful in defining other SQL/DS tables or views that the user can access. If this requirement is not met, the related section of the tree view within the "Host SQL Query" dialog cannot be used.

Default values:

SYSCATALOG: "SYSTEM.SYSCATALOG"
SYSCOLUMNS: "SYSTEM.SYSCOLUMNS"
SYSINDEXES: "SYSTEM.SYSINDEXES"

Automatic display of table indexes in SQL tree view

The "Host SQL query" tree view (see page 160) allows to retreive the columns of a table. When activating option "Automatic display of table

indexes in SQL tree view", index information will be shown additionally. If this is not wanted (e.g. for performance reasons), you can get these info by selecting the appropiate context menu entry.

Host statistics shall include data directory access

Host directory accesses only will be included in the host statistics when setting this option. Additionally, the statistic function has to be enabled on the host.

Max structured host download IO's having no relevance for result

This field defines the maximum number of read operations to be performed for structured host downloads. If this number is reached and no records have been selected yet, the data transmission is terminated and an error message displays.

2.2.4.3.6 Various

This dialog contains timeout and data transmission settings.

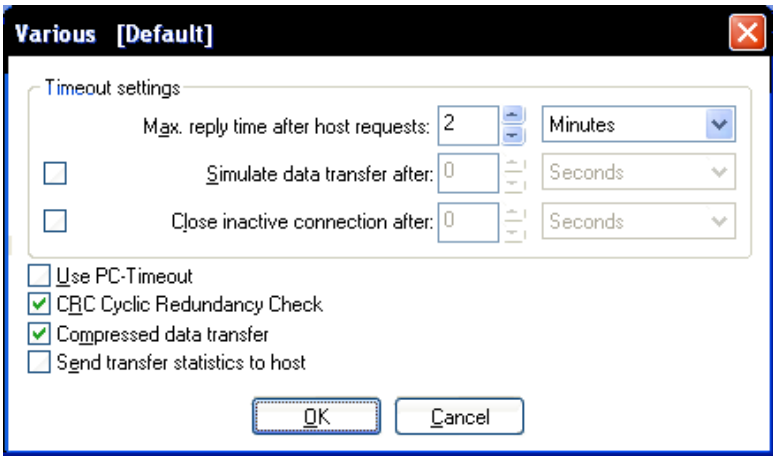


Fig. 119: Dialog "Various"

The dialog components are described below:

Max. reply time after host request

This input field contains the longest possible response time expected from the mainframe. If the host does not answer a request within the specified interval, a HOST TIMEOUT ERROR is returned to the PC application. The timeout can be a value between 1 and 20000 seconds.

Simulate data transfer after

Some host monitors or session managers on the host terminate open connections after a certain period of time without any keyboard input. This is not an acceptable practice for tcACCESS LU2 connections. Therefore, a time interval in minutes may be specified in this field. When the defined time has passed, the PC Communication Handler simulates a user input and sends it to the host. If this field contains the value '0', the function is inactive.

Close inactive connection after

This setting specifies, that after the defined time interval the tcACCESS Communication Handler should automatically close the inactive connection.

Note to "Connection Pooling":

Using this timeout has the effect of a so called "Connection Pooling". The default setting defines the name of a default configuration file and not a particular connection slot. This setting results in the following:

If a user establishes a session, a free slot is being used for the desired configuration. This connection then will be used. If a second connection is needed, the first connection will be used, if this connection does not currently support an active request. Otherwise a new connection will be established. This leads into a Pool of multiple connections. If a connection is inactive for a defined time interval ("Close inactive connection after"), it will be automatically closed.

Use PC-Timeout

When you select this option, the transfer timeout (option "Max. reply time after host request") will be controlled by the tcACCESS client. If this option is not active, the tcACCESS host component controls the transfer timeout.

CRC Cyclic Redundancy Check

When you select this option, a 32-bit CRC check sum is built in order to verify the data transfer between local PC and host. Thus, transmission errors are recognized in almost any case.

Compressed data transfer

If you select this, the data is compressed before it is sent and decompressed accordingly at the receiver. This option is helpful when the following conditions exist:

- When the data to transfer is suitable for compression (e.g., texts).
- On a slow communication line between PC and host.
- On a fast local PC (today, this should no longer present a problem).

Send statistic to host

Selecting this option will cause the tcACCESS data transfer to send a stattistic record to the host after the transfer.

2.2.4.4 Connection Slots

Each data transmission between the PC and the mainframe uses a slot. To activate a slot, you must assign a connection configuration to the slot. Any number of slots may be assigned to a configuration.

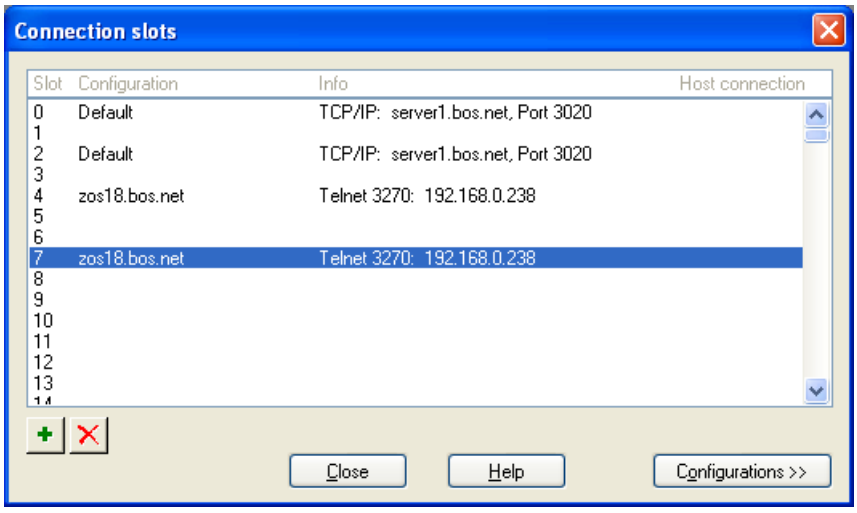


Fig. 120: Dialog "Connection slots"

The dialog components are described below:

Slot-List

To use a slot for data communication you must assign a configuration using this list. Multiple lines can be processed and edited. The list displays information about the connection parameter for each slot. A "+" sign will be displayed at the end of the line, if the slot is currently in use.

The icons are described below:

Assing a connection configuration to all selected slots.

Delete the connection configuration from all selected slots.

Close

Close this dialog.

Help

Display the online help screen(s).

Configurations >>

Clicking on this button will toggle directly to the Host Connections page 192.

2.2.4.5 tcACCESS Host Security Administration

This dialog is only available when a PC user has performed a valid administrator signon (see Administrator Logon on page *186*) and when a host session is established. This function is not available under the VM host operating system. If it is not possible to use the tcACCESS Host Security Administration for any reason, the related menu entry is displayed in grey.

Note:

- All users and authorizations defined here are only relevant if tcACCESS is configured on the host to use the internal tcACCESS host security. If the security is handled via a RACF database, this dialog cannot be used for security administration.

- With its integrated tcACCESS host security, tcACCESS offers various possibilities to restrict selected users from accessing certain host resources.

- For identification, the user name CICS USERID or OPID is used, which was entered when the connection was established. The Host Communication Handler saves this identification and passes it to all tcACCESS Host Server Modules.

•

The following is an example of the tcACCESS Host Security Administration dialog:

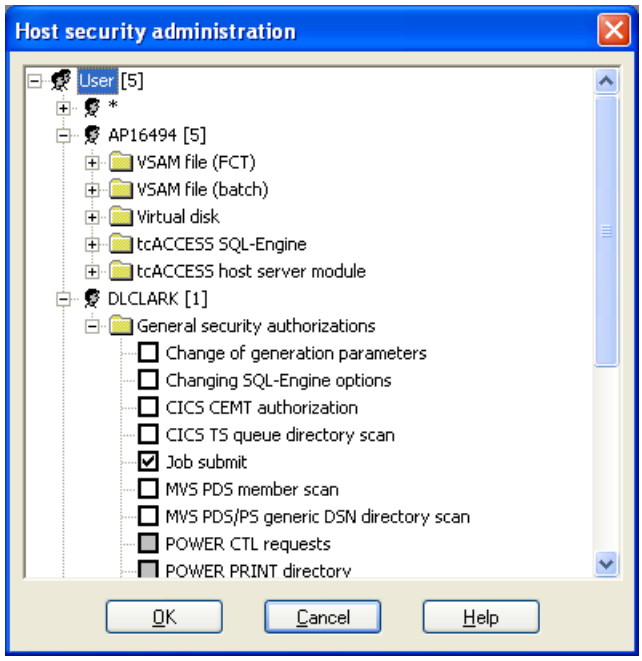


Fig. 121: tcACCESS Host Security Administration

The dialog components are described below:

Tree view

The main part of the dialog is the security tree view. When the dialog is open, it displays the users defined within the tcACCESS host security.

OK

After selecting this button, all modified rights are transferred to the mainframe. However, all deleted or newly added users are removed or added directly during the related operation on the host.

Cancel

After selecting this button, the dialog closes without saving any changes. However, the deleted or newly added users are removed or added directly during the related operation on the host. In other words, these changes take effect.

2.2.4.5.1 Adding new users to tcACCESS Host Security

There are two ways to define a new user for the tcACCESS host security:

1. Right-click **Add new user** menu entry for the first tree view node (**User**).

   The appropriate dialog displays and the name for the new user can be entered into the NEW USER field. If an existing user is selected from the USER TO BE COPIED list, the new user is assigned the same security access profiles and the identical password. However, if it remains empty, the new user has no security access profiles and no password immediately after creation.

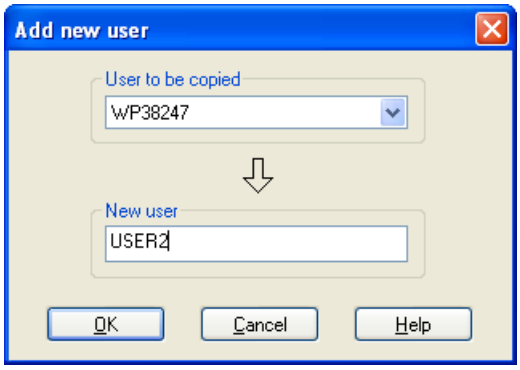


Fig. 122: tcACCESS Host Security Administration - Dialog "Add new user"

2. Right-click **Copy user** menu entry for an existing user entry.

   A user is added with the name entered in the NEW USER FIELD. The new user entry is initialized with the authorization rights and the password of the copied user.

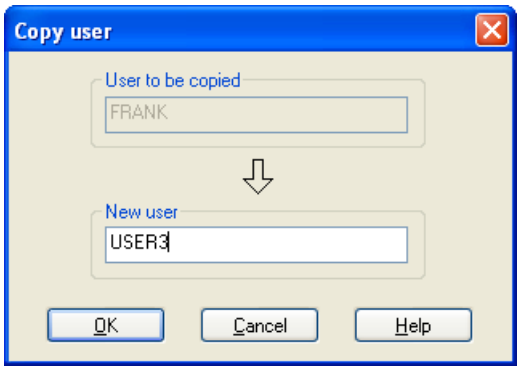


Fig. 123: tcACCESS Host Security Administration - Dialog "Copy user"

The default password of the new user is the same as the user name. An appropriate entry is made in the tcACCESS host security file at the same time the user entry is added to the tree view. The update remains constant when the

tcACCESS Host Security Administration dialog is terminated by selecting the **Cancel** button.

2.2.4.5.2 Deleting users from the tcACCESS Host Security

You can remove a user entry by selecting the DELETE USER menu item. Right-click on the entry to access this menu selection.
A corresponding entry is made in the tcACCESS host security file at the same time the user entry is deleted from the tree view. The update remains constant when you select the **Cancel** button to terminate the tcACCESS Host Security Administration dialog.

2.2.4.5.3 Defining a password for a user entry of the tcACCESS Host Security

If the tcACCESS host security also applies to user administration based on the tcACCESS mainframe configuration, it is necessary to define a password for each user entry.

For this, use the CHANGE PASSWORD FOR tcACCESS HOST SECURITY menu item. Right-click the user entry in the tree view to activate this function. A dialog displays and allows you to enter a new password for the selected user.

Security"

Fig. 124: tcACCESS Host Security Administration - "Change password for tcACCESS Host

The password can be up to 8 characters long and may contain numbers and letters (umlauts are accepted).

When a user entry is newly created, no password is assigned. If an existing user entry is copied, the password of the copied user entry is assigned.

2.2.4.5.4 Converting a resource access profile of a tcACCESS Version older than 3.0

If a user is addressed in the tree view of the tcACCESS Host Security Administration and the authorizations of this user have been created with a tcACCESS version older than 3.0, the CONVERT OLD RESOURCE ACCESS PROFILES dialog displays.

Since in previous tcACCESS versions there were no *type identifiers* for authorization labels, you should assign one or more security types to the labels shown in the list of OLD RESOURCE ACCESS PROFILES. If this dialog is ended with **Cancel** or if there are still old resource access profiles that do not have at

least one type assigned, it is impossible to continue processing for the related user.

profiles"

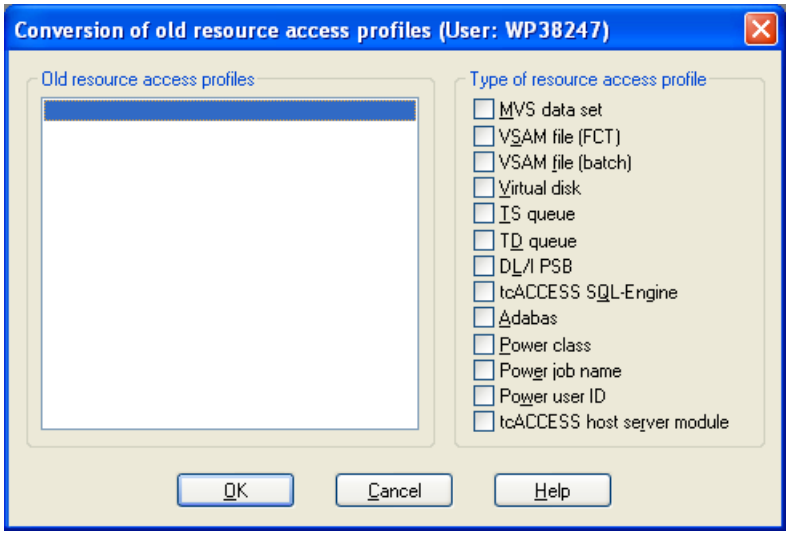


Fig. 125: tcACCESS Host Security Administration - Dialog "Conversion of old resource access

An entry is only made into the related tcACCESS host security file when you click the **OK** button to close the tcACCESS Host Security Administration dialog.

2.2.4.5.5 Inheriting security settings from a user

If several users are to have the same authorizations assigned, it is not necessary to define them for each individual user. You can define authorizations for one of the users and then use the inheritance function for all other users. You can activate this function via the context menu item (clicking the right mouse key) INHERITANCE OF SECURITY SETTINGS for a user entry within the tree view. The following dialog displays:

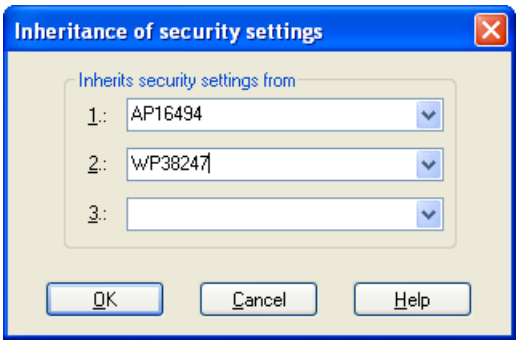


Fig. 126: tcACCESS Host Security Administration - Dialog "Inheritance of security settings"

You can select up to three users whose security settings are to be inherited.

This function makes it possible to administrate and assign groups of authorizations using *a logic user*. For example, the logic users VSAMUSER and SQLUSER may be defined with the desired authorizations; then a *real user* can

be assigned the same authorizations by using the INHERITANCE OF SECURITY SETTINGS function.

Another advantage of this method is that any requested changes regarding the authorization can be performed for a single user entry only and all other users are updated automatically due to the inheritance function. The entry is only added to the related host security file when the tcACCESS Host Security Administration dialog is closed by selecting **OK**.

2.2.4.5.6 Authorization types

tcACCESS supports a multitude of authorizations for host access. The verification of these authorizations is performed at runtime on the mainframe.

In combination with the use of placeholders ("*"), resource access profiles can be applied to many of the existing system resources (e.g., 'MYFIL*' includes all files with 'MYFIL' at the beginning of their name). When authorizations are verified, a completely defined resource name (e.g., 'MYFIL12') has a higher priority than one with placeholders (e.g., 'MYFIL*'), i.e., only the first of both entries is applied.

It is possible to use several place holders "*" as shown in the following examples:

- QUAL1.*.*.DATA
- QUAL1.*QUALX*

The following authorization types are available:

General security authorizations

This type includes all authorizations that are either admitted or refused. A detailed specification is not necessary.

The following authorizations belong to this group:

- CICS CEMT authorization
- CICS TS queue directory scan
- Job submit
- MVS PDS/PS generic DSN directory scan
- MVS PDS member scan
- Print/punch data receive
- Print/punch data submit
- SQL database access
- VSAM files (batch) directory scan
- VSAM files (FCT) directory scan
- POWER CTL requests
- POWER PRT directory
- POWER PUN directory
- POWER RDR directory

MVS data set

The MVS data sets to be considered must be specified with their DSN names.

Examples:

- SYS1.MACLIB
- USER35.*.DATA

VSAM file (batch)

For batch VSAM files the name must be entered.

Examples:

- VSE.CONTROL.FILE
- VSE.*

VSAM file (FCT)

For internal VSAM files the name of their CICS FCT definition must be entered.

Examples:

- VSAMF1
- VSAM*

Virtual disk

Virtual disks must be specified in the following format:

BaseHostFile.Partition

BaseHostFile is the CICS FCT name of a VSAM file that contains a virtual disk in each partition.

Examples:

- TCAVDI.DISK1
- TCAVDI.SYS*
-

It is also possible to individually secure the logic files and directories within a virtual disk. For this use the following specification is to be made:

BaseHostFile.Partition:sub-directory1/sub-directory2/.../file_name

Examples:

- TCAVDI.SYS:SUBD1/SUBD2/FILE1
- TCAVDI.SYS:SUBD1/SUBD2/*

TS queue

TEMPORARY STORAGE resources must be specified with their TS queue name as defined under CICS.

Examples:

- QUEUE
- QUE*

TD queue

TRANSIENT DATA resources must be specified with their TD queue name as defined under CICS.

Examples:

- QUEUE
- QUE*

DL/I PSB

In order to restrict DL/I access, it is necessary to enter the PSB name, which is used under CICS.

Examples:

- PSB1
- PSB*

tcACCESS SQL-Engine

Tables and views of the SQL-Engine must be defined with their complete name (Creator.Name).

Examples:

- SYSTEM.SYSTAB
- USER1.USERT*

Adabas

Adabas files must be defined in the following format:

dddfff

(*ddd* stands for the 3-digit database number and *fff* for the 3-digit file number.)

Examples:

- 008005

- 008006

If the PREDICT file names are to be verified due to the generation of the host server module TCASADAQ, it is necessary to enter the resource name.

Example:

- EMPLOYEES

(The *Read* attribute can only be defined for ADABAS files.)

Power class

This authorization is only applied when an appropriateTCAPARM configuration is given. The desired POWER JOB CLASS is entered. Only the access types *Read* and *Write* are used.

Examples:

- A
- *

Power job name

This authorization is only applied when an appropriate TCAPARM configuration is given. The desired POWER JOB NAME is to be entered. Only the access types *Read* and *Write* are used.

Examples:

- LST1
- LST*

Power user ID

This authorization is only applied when an appropriate TCAPARM configuration is given. The desired POWER USER ID is to be entered. Only the access types *Read* and *Write* are used.

Examples:

- USER1
- USER*

tcACCESS host server module

This authorization allows you to restrict the use of certain tcACCESS host server modules. It is only verified when it is activated on the tcACCESS host component.

Examples:

- TCASADAQ
- TCASA*

tcACCESS load module

This authorization allows you to restrict the download of certain load modules.

Examples:

- DFHSAM04
- DFHIV*

VSE Library

This authorization allows you to restrict the access to modules in a VSE library.

Examples:

- LIB.SUBLIB
- DLI.*

2.2.4.5.7 Granting/revoking general security authorizations

In order to grant a general security authorization (see Authorization types, page *211*), select the related sub-entry from the context menu item (clicking the right mouse key) ADD RESOURCE ACCESS PROFILE for a user entry on the tree view. This results in all general security authorizations displays in the tree view. They can be either activated on request (✓ in front of an entry) or removed (no ✓ in front of an entry).

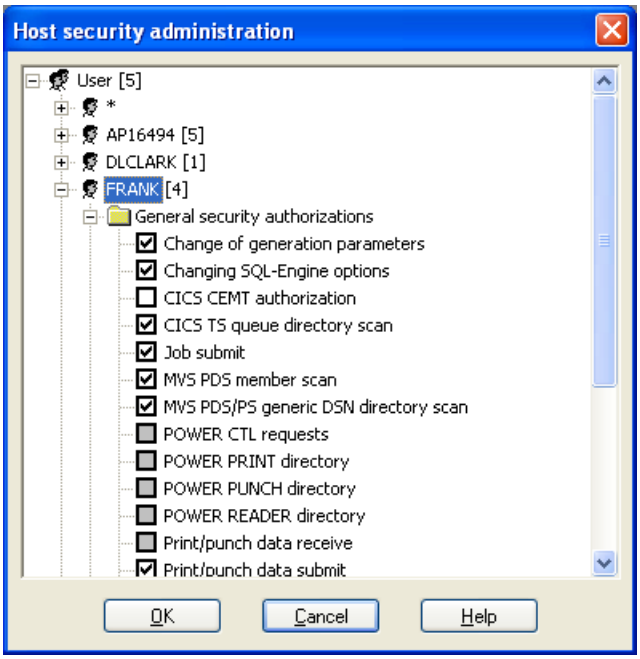


Fig. 127: tcACCESS Host Security Administration - Grant general security authorization

When at least one general security authorization is active, the related sub-tree displays in the tree view and allows you to open it directly to perform any changes.

An entry to the related host security file is only added when you select the **OK** button to close the tcACCESS Host Security Administration dialog.

2.2.4.5.8 Adding resource access profiles

There are two methods for adding a resource access profile (see Authorization types, *page 211*), which is not part of the general security authorizations:

1. Select sub-item on the context menu (right mouse key) ADD RESOURCE ACCESS PROFILE for a user entry on the tree view. The appropriate dialog displays to enter the preferred resource name.

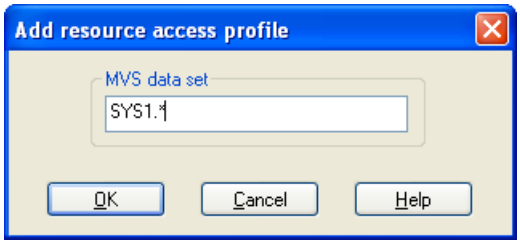


Fig. 128: tcACCESS Host Security Administration - Dialog "Add resource access profile"

2. Select the context item (right mouse key) COPY RESOURCE ACCESS PROFILE for an authorization listed in the tree view. A dialog opens and allows you to enter the preferred new resource profile. All access types permitted for the copied resource are effective for the new entry.

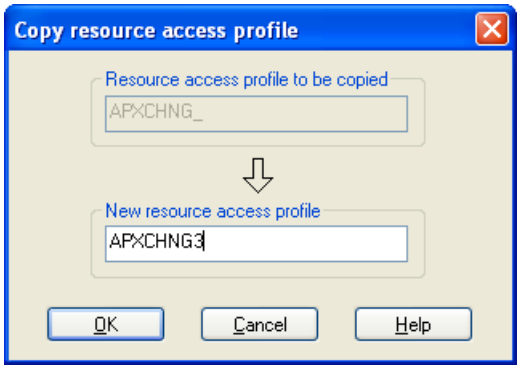


Fig. 129: tcACCESS Host Security Administration - Dialog "Copy resource access profile"

Independent of how it is created, the new resource access profile displays in the tree view with opened sub-entries. They allow you to define the permitted access types for the created resource entry. If a '✓' is shown in the box on the left of an entry, the access type is active.

For most authorizations, the following five options of access types are available (sometimes less):

- **Read -** Read access
- **Browse -** Sequential read
- **Write -** Write (add) access
- **Update -** Write (alter) access
- **Delete -** Remove access

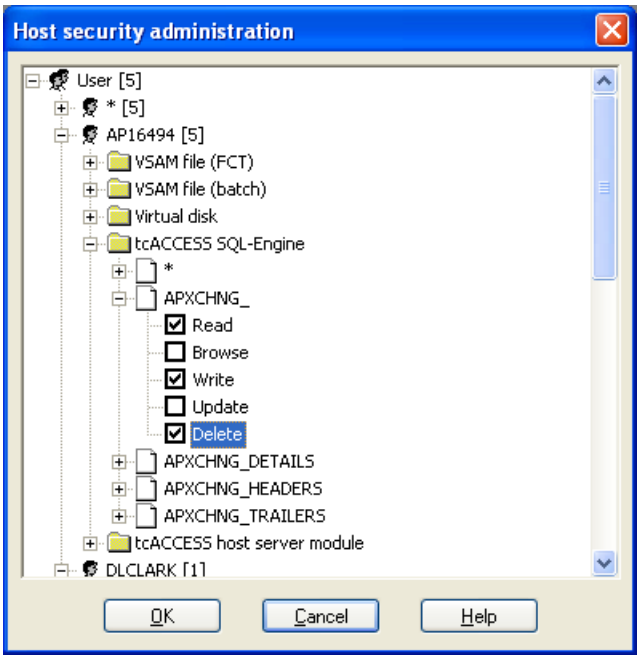


Fig. 130: tcACCESS Host Security Administration - Selection of granted access types

A registration within the related host security file is only made when the tcACCESS Host Security Administration dialog is closed by selecting **OK**.

2.2.4.5.9 Deleting resource access profiles

Use the context menu item (right mouse key) DELETE RESOURCE ACCESS PROFILE for an entry in the tree view can be used to remove an authorization that does not belong to the GENERAL SECURITY AUTHORIZATIONS group.

A registration within the related host security file is only made when the tcACCESS Host Security Administration dialog is closed by selecting **OK**.

2.2.4.6 tcACCESS-Wizards

In oder to use the tcACCESS Wizards the user must sign on as an administrator (refer to Administrator Logon, page 186). The Wizards assist you in the creation of APPC connections and help you to dynamically change tcACCESS system-generation parameter.

2.2.4.6.1 APPC-Wizard

The APPC Wizard eases the process of defining APPC connections of a Batch-region or partition to an online-system. The wizard will prompt you for the various parameter and leads you through a step-by-step dialog to define the necessary VTAM-definitions, the definitions required by the online-system and the definitions for the tcACCESS system-generation. All individual steps will be commented. This step-by-step explanation can also be printed.

2.2.4.6.2 Generation Wizard

This wizard can be used to maintain and change the tcACCESS system-generation parameter. All parameter that can be changed will be retrieved with their current value and a short explanation of the parameter. The parameter can be changed interactively and can be saved to the Virtual Disk.

2.2.4.6.2.1 Using the Generation Wizard

When starting the Generation Wizard all generation parameter, that are currently being used will be retrieved from the host. The Generation Wizard can only be started if there is an active host connection.

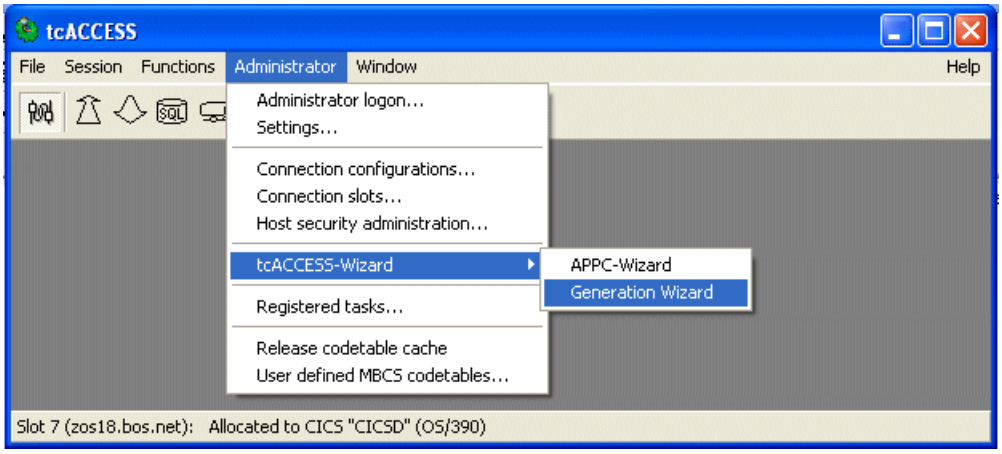


Fig. 131:Generation Wizard – Start the Generation Wizard

Next, a welcome screen will be displayed that gives a brief description about the various functions of the Generation Wizard.

2.2.4.6.2.2 Display the generation parameter

Click on the "Next"-button and a dialog displays that shows the generation parameter and the current values. A brief description and the default values will be displayed for every parameter.

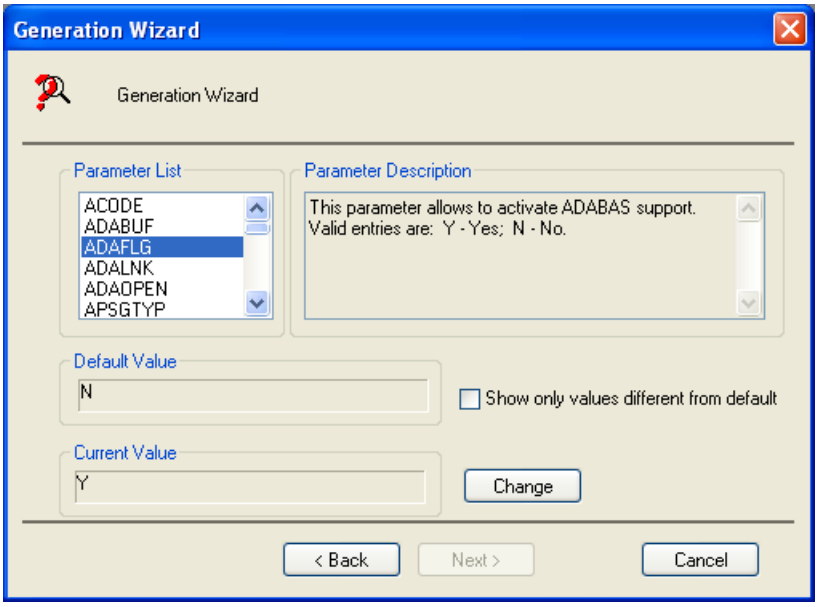


Fig. 132: Generation Wizard – Display of parameter

The dialog components have the following meaning:

Parameter list

All available tcACCESS generation parameter will be displayed in alphabetical order. Selecting a parameter will display the current setting, the default value and a short desvription.

Parameter description

A short description of the parameter will be displayed in this field. You will be informed about the settings you can choose and about their results.

Default value

Display of the default value for the selected generation parameter. This value will be used if no other value has been specified.

Current value

Display of the current value of the selected generation parameter.

Show only values different from default

If this option is active, only the parameter will be displayed whose current value differs from the default value.

Change

Click this button, if you want to temporarily or permanently change a parameter value. A dialog opens where you can change the setting of the selected parameter. Some parameter can only be changed by performing a host generation job. For those parameter the "Change"-button is inactive.

2.2.4.6.2.3 Maintain generation parameter

This dialog allows the specification of a new value of the selected generation parameter. You can test the new setting and the new parameter will be temporarily saved on the host. Alternatively the changes can be permanently saved.

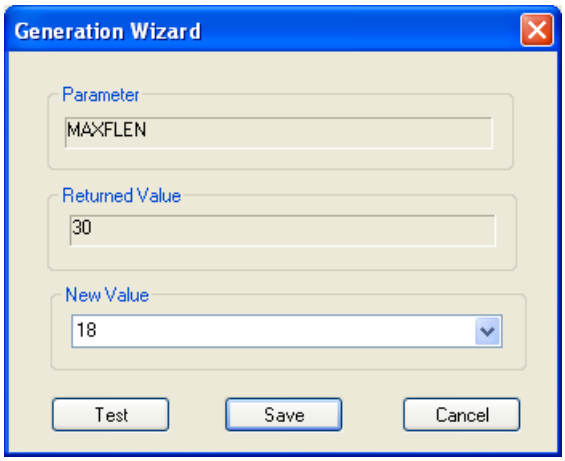


Fig.. 133: Generation Wizard –Maintain generation parameter

The dialog components have the following meaning:

Parameter

Display the generation parameter whose value should be changed.

Returned value

Display of the current setting for the generation parameter on the host.

New value

Specify the new value for the generation parameter. There is no syntax check for the new value. You can use the "Test"-button to check the new value.

Test

This button temporarily saves the new value on the host. The new value will be sent to the host and remains active as long as the value is not being overwritten by another value or until a restart of tcACCESS.

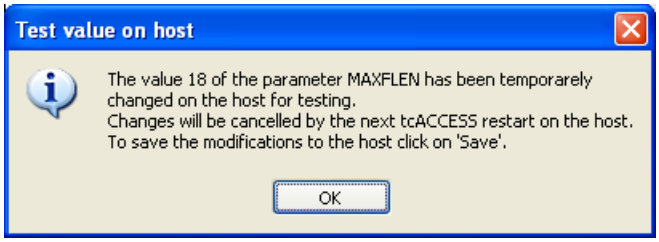


Fig. 134: Generation Wizard –Notification about temporary save

Save

The "Save"-button will permanently save the changes to the generation parameter. The new value will be saved to a temporary file on the PC and the dialog will be closed. Any number of parameter can be changed this way. All the changes are saved to the same temporary file on the PC. The changes can then be transferred to the host all at once. The actual transfer to the host will be performed in another step of the Generation Wizard.

2.2.4.6.2.4 Save generation parameter

After having maintained the generation parameter, clicking on the "Next"-button will lead you to step "Save as" of the Generation Wizard. This dialog asks for the name of a file on the Virtual Disk. All generation parameter and their values, that are different from the default values, will be saved. The values stored in this file will be loaded by tcACCESS at initialization time and the original values, even the ones that have been generated using the generation job, will be overwritten by the new parameter.

The override file on the Virtual Disk will only be loaded by tcACCESS if the parameter "INIFILE" specifies the name of the file. Processing sequence is as follows: During the first call of tcACCESS the main directory of the Virtual Disk specified in parameter 'VDISK' will be searched for this file. The values contained in this file will be used. If the generation parameter "INIFILE" has been set to 'NO' the file is not being searched for, meaning that the changes performed by the Generation Wizard will NOT be applied.

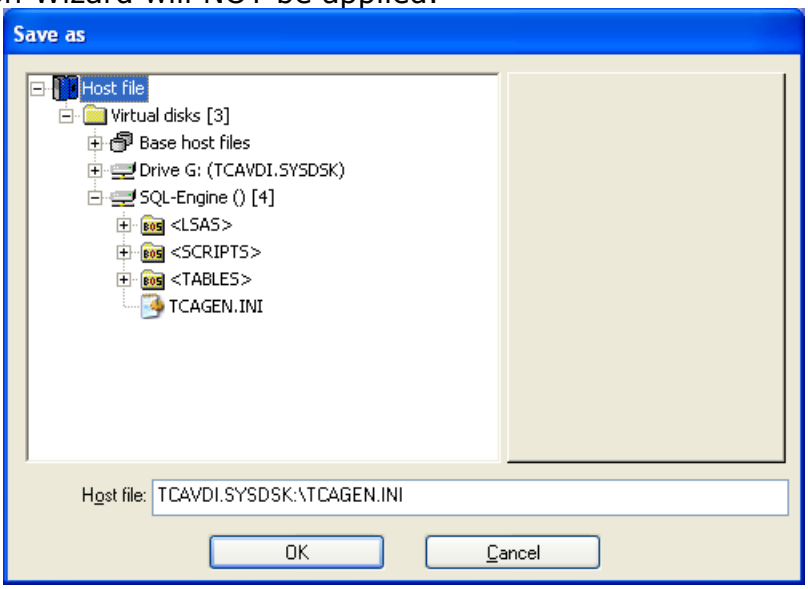


Fig. 135: Generation Wizard –Save as

Click "OK" and the file is being transferred to the host and saved to the Virtual Disk. A message box will be displayed informing you that the changes will become effective after a restart of tcACCESS. The Generation Wizard terminates.

2.2.4.7 Registered tasks

If the user has signed-on as administrator (refer to Administrator Logon, *page* 186) it is possible to display the tasks that are currently registered on the connected host. In addition the dialog can be used to start tasks from the PC or to cancel a currently running task on the host.

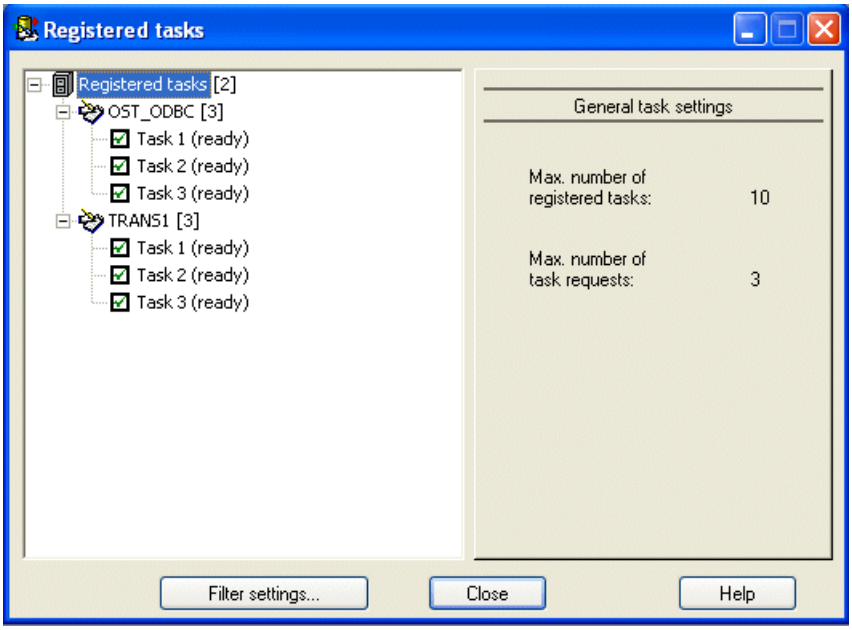


Fig. 136: Registered tasks - treeview

The dialog components have the following meaning:

Treeview

The treeview displays all tcACCESS listener tasks that are registered on the mainframe. The task registrations may have been performed from different workstation clients (tcACCESS listener installations on different computers/platforms).

Tree-entry "Registered tasks":

Click on the plus-sign in front of the tree-entry the tasks that are registerd on the connected mainframe will be retrieved and displayed. The number of registered tasks is displayed in [ ]. The right part of the screen dislays a window with basic taks settings. The values of the host generation parameter "Maximum number of registerd tasks" and "Maximum number of task requests" are displayed.

If you right mouse-click on a task entry the following context menu is displayed:

Update

Refresh the list of registered tasks.

Filter settings ...

Opens the dialog "Filter Settings"

Display all tasks

All tasks are being displayed in the treeview. Filter settings will not be applied

Apply filter

If filter settings have been defined they will be applied on the treeview. Only these tasks will be displayed that match the filter criteria (i.e. all tasks that have been registered from a certain workstation).

Tree-entry "Task name":

Selecting a task in the treeview displays various task information in the window right to the treeview (titled "Registered Tasks"). The information include workstation name, connection type, TCP/IP-address etc.
A click on the plus-sign in front of the task displays the "**Task requests".** The number of entries corresponds to the definition of "Maximum number of task requests" in the host system generation.. Tasks with a green check-mark are "ready to go". Tasks with a red check-mark have been started and are currently in execution.

If you right mouse-click on a task entry the following context menu is displayed:

Update

Refresh the registered tasks and corresponding task requests.

Filter settings ...

Opens the dialog "Filter Settings"

Display only this task

Only the selected task will be displayed.

Display tasks with this PC name

All tasks will be displayed in the treeview that have been registered by the same workstation than the one that is currently selected.

Display all tasks

All tasks will be displayed in the treeview. No filter is applied.

Apply filter

Settings made in dialog "Filter settings" will be applied to the treeview, hence only tasks are displayed that match the filter settings (i.e. display all tasks that have been registered by a certain workstation).

Cancel requests

The processing of the task on the host will be cancelled. The task will be registered again and the task requests will be reset.

Deregister task

Delete the task registration at the host.
Note: The deregistration of the task is only performed on the host. The workstation listener that has originally registered the task will not

be informed. During the test of "validity of host task registration" the task will be registered again by the workstation listener..

Start task

This opens dialog "Start task..." that lets you start tasks that are registered on the host. All necessary commands and parameter will be sent to the corresponding host server-module according to the settings in the dialog. The host module will then start the task.

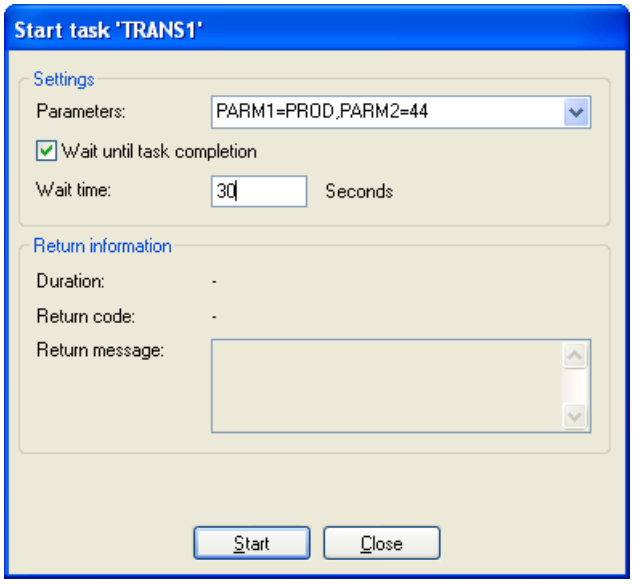


Fig. 137: Dialog "Start task"

The dialog components have the following meaning:

Parameter

Specify the parameter that must be passed to the task in the format of PARAMETERNAME=VALUE. The parameter must be supplied as comma separated list.. These parameter are being passed to the program defined in the task definition.

Wait until task completion

When this option is active a wait is issued for the specified period on the completion of the task. After completion of the task any returncode or return-message will be displayed in the dialog.. If this option is not active, the dialog will be immediately closed after the task has been started.

Wait time

Specify the maximum time to wait for the completion of the task. If nothing has been specified the waiting for task completion will be unlimited.

Duration

Display the duration time of the task.

Return code

Display the returncode of the task.

Return message

If the task has returned a message, it will be displayed here.

Start

Use this button to start the task will all settings and parameter defined in the dialog.

Close

Close the dialog.

Tree-entry "Task request":

If a "Task request" of a task has been selected information about the task request will be displayed in the window titled "**Task request**". As an example the information include task status, last start, last end or return message.

Performing a right mouse-click on a "Task request" opens a context menu. The menu entries are identical to the ones described above for the "Task" context menu.

Close

Closes the dialog "Registered tasks"

Filter settings

This button opens dialog "Filter settings". The defined filter conditions are being applied to the treeview of the registered tasks. Example: If a workstation-name of "SERVER1" has been defined only tasks that have been registered by workstaion "SERVER1" will be shown with option "Apply filter".

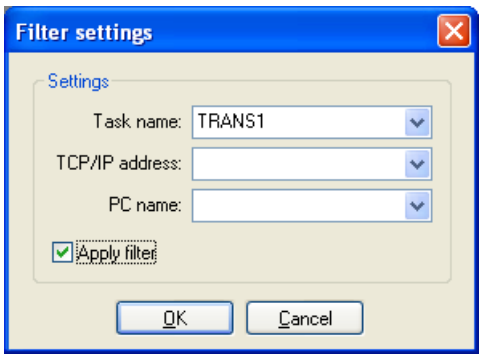


Fig. 138: Filter settings

The dialog components have the following meaning:

Task name

Name of the task that should be displayed in the treeview.

TCP/IP address

Only tasks that will be started using the defined TCP/IP address will become part of the treeview.

PC name

Specifies the name of the workstation that is responsible for the task registration. Only tasks registered by this workstation will become part of the treeview display.

Apply filter

This option activates the filter.

OK

The filter settings will be saved. If option "Apply filter" has been set, the filter immediately becomes effective.

Cancel

Close the dialog. No changes take effect.

2.2.4.8 Release Codetable Cache

Codepages used by queries are kept in storage for immediate access. Using this menu entry a user who has signed on to tcACCESS as an administrator (refer to Administrator Signon) may release the storage that has been occupied by the "Global Language Pack". The next query that requires a codetable performs a new load of the table.

2.2.4.9 User defined MBCS Codetables...

Using this menu entry a user who has signed on to tcACCESS as an administrator (refer to Administrator Signon) can view the MBCS codetables on the mainframe and can edit them or even create new codetables.

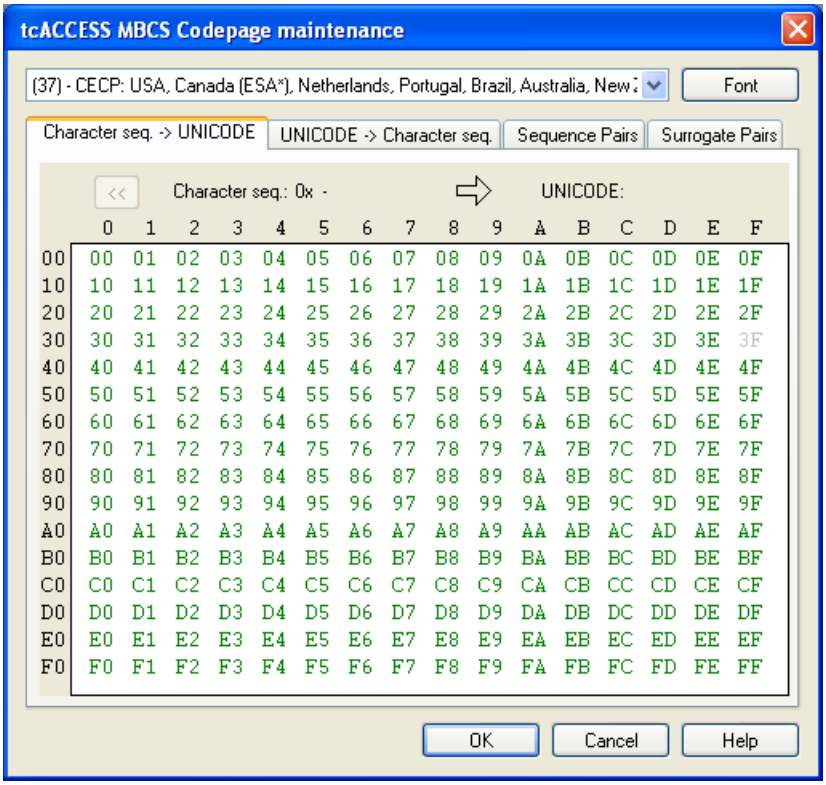


Fig. 139: MBCS Codepage maintenance

The dialog components have the following meaning:

Selection list of available MBCS Codepages

The selection list displays all codepages that are available on the mainframe. This includes standard- and user defined codepages. The user defined codepages are at the end of the list and are marked with an asterisk (*). Entry "New user defined table" can be used to create a new user defined table (refer to New user defined table, page 242).

Tabs

Use the tabs to view the different conversions and – for user defined codepages – modify them. The different tabs are explained in detail in this chapter.

OK

Closes the dialog and apllies the changes.

Cancel

Closes the dialog but no changes will be applied. A confirmation box will be displayed.

2.2.4.9.1 Tab "Character seq. -> UNICODE"

This tab displays the current conversions of character sequeneces to UNICODE. For user defined codepages they can also be modified.

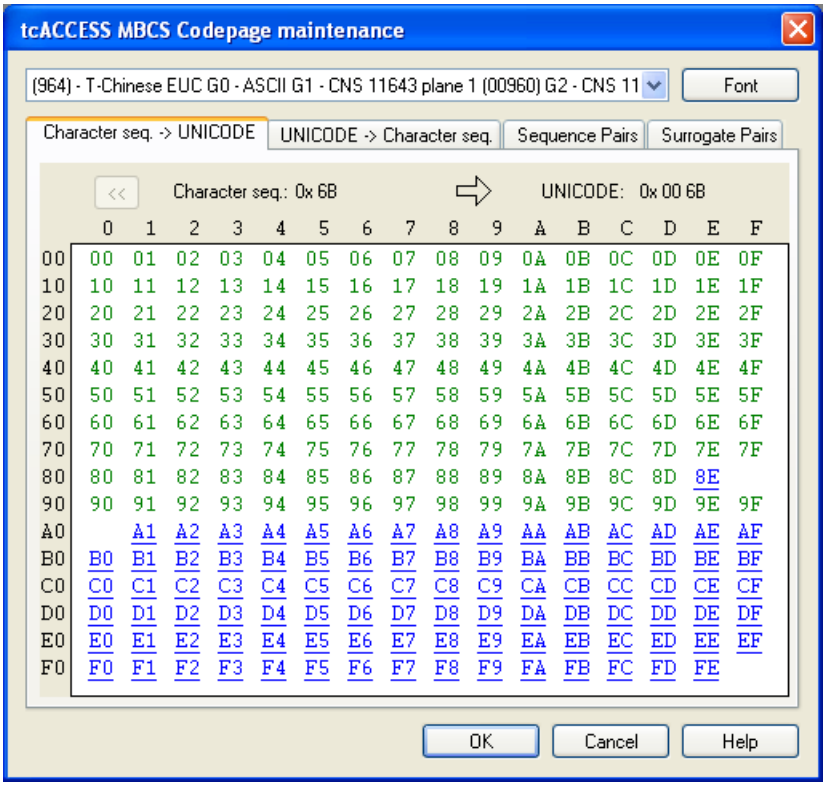


Fig. 140: Tab "Character seq. -> UNICODE"

The dialog components have the following meaning:

Conversion table

This table shows all existing and possible conversions from a character sequence to the corresponding UNICODE value.

Moving the mouse pointer across the entries the corresponding conversions are being displayed in the upper part of the table.

A right mouse click displays the converted UNICODE character sequences. Another right mouse clcik toggles back to the previous display.

A left mouse click on an entry of a user defined table can result in an action. Possible actions and the meaning of the table entries are:

no entry: A conversion of this character sequence into a UNICODE value is not allowed (i.e. the gap in figure 228 between 0x8E and 0x90).

Green entries: A conversion of this character sequence into a UNICODE value exists (i.e. The conversion from 0x6B to 0x006B in the above figure or 0x8EA2F2C4 to 0x9F98 in the following figure). A left mouse click displays a dialog that allows to change the conversion (refer to Change codetable entry, page 236).

Green entries: A conversion of this character sequence into a UNICODE Sequence or Surrogate Pair exists (i.e. The conversion of 0xB342 to 0x02000B in figure xxx). A click with the left mouse key changes to tab "Sequence Pairs" or "Surrogate Pairs" and the selected entry will be marked.

Blue entries: This value is only part of a longer character sequence (i.e. 0x8E in the above figure). Using this mechanism a character sequence can be built (based upon a codepage) that may consist of up to 4 single values (Bytes, i.e. 0x8EA2F2C4 in figure 133). With a left mouse click the next level of the characterstring is displayed (all character string conversions that consist of blue values that have already been clicked on).

Grey entries: No conversion existst of this character sequence into a UNICODE value although a conversion is basically allowed (i.e. Character sequence 0x8EA2F2C5 in figure xxx has no UNICODE value assigned). A left mouse click displays a dialog that allows the creation of a conversion (refer to Change codetable entry, page 236).

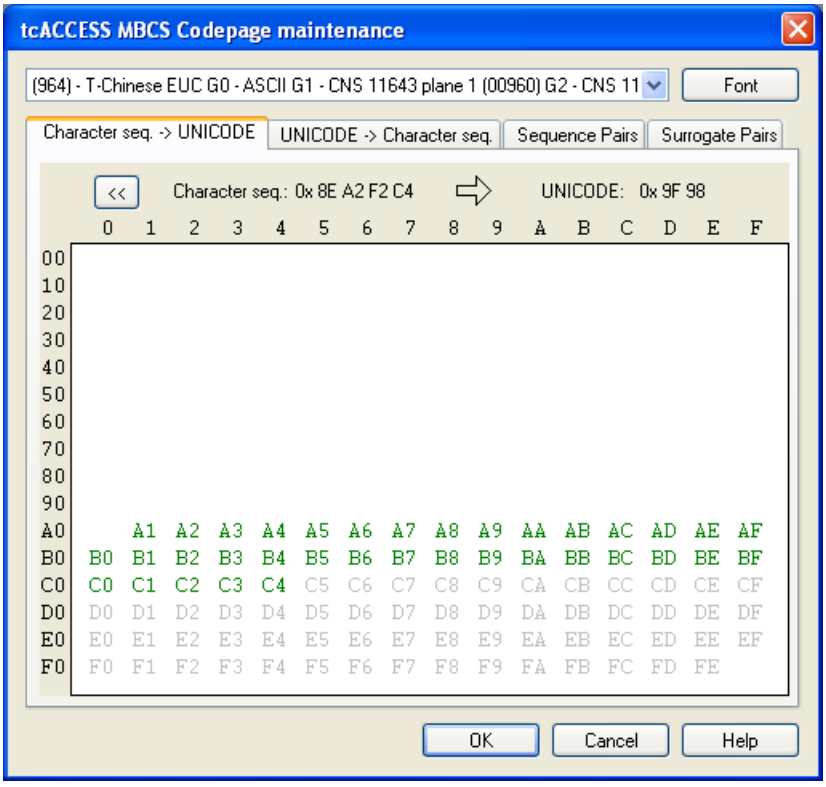


Fig. 141: Tab "Character sequence -> UNICODE" with composite character sequence

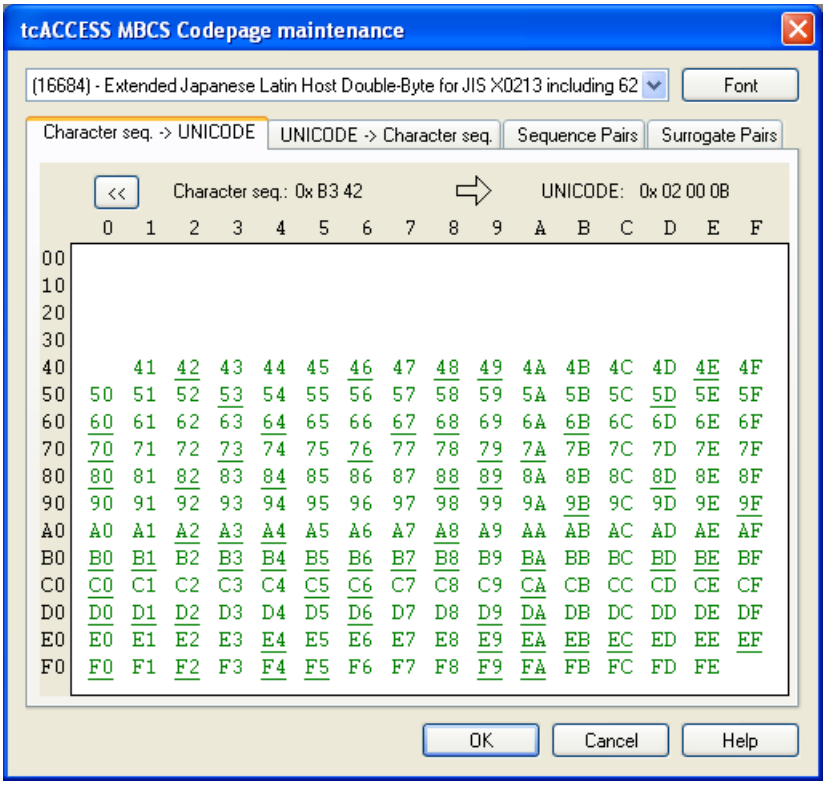


Fig. 142: Tab "Character sequence -> UNICODE" with Sequence or Surrogate Pairs

Button "<<"

Pressing this button goes back one level in the character sequence (i.e. from 0x8EA2F2 to 0x8EA2 in figure 141). This button is only active if the level of the character sequence has already be increased because a blue entriy has been pressed.

Buttons "SI" and "SO"

Some codepages have the special attribute of a "Stateful-EBCDIC"-coding. This means that they can convert one byte long character sequences as well a character sequences that are two bytes long into UNICODE. So called "Shift In" and "Shift Out" characters of these codepages toggle between the two different states. There are two conversions for the first level of the character sequence. Pressing button "SI" and "SO" toggles between these two levels and the text on the button changes (i.e. the 1 byte conversion in figure 142 and the 2 byte conversion in figure 143). The buttons are only active for a "Stateful-EBCDIC" codepage.

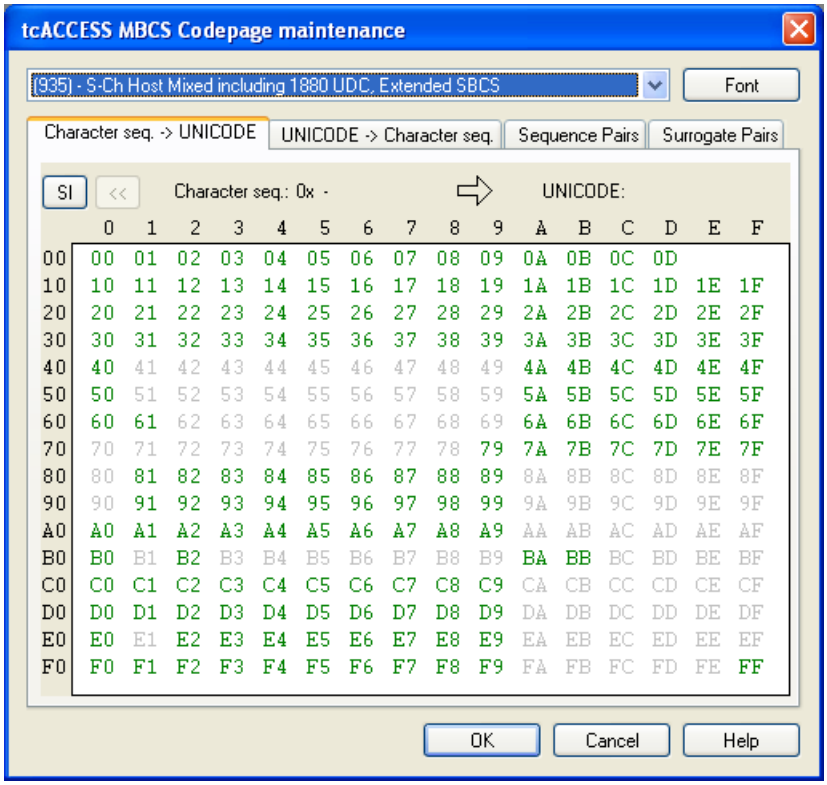


Fig. 143: Tab"Character seq. -> UNICODE" with "SI"-button

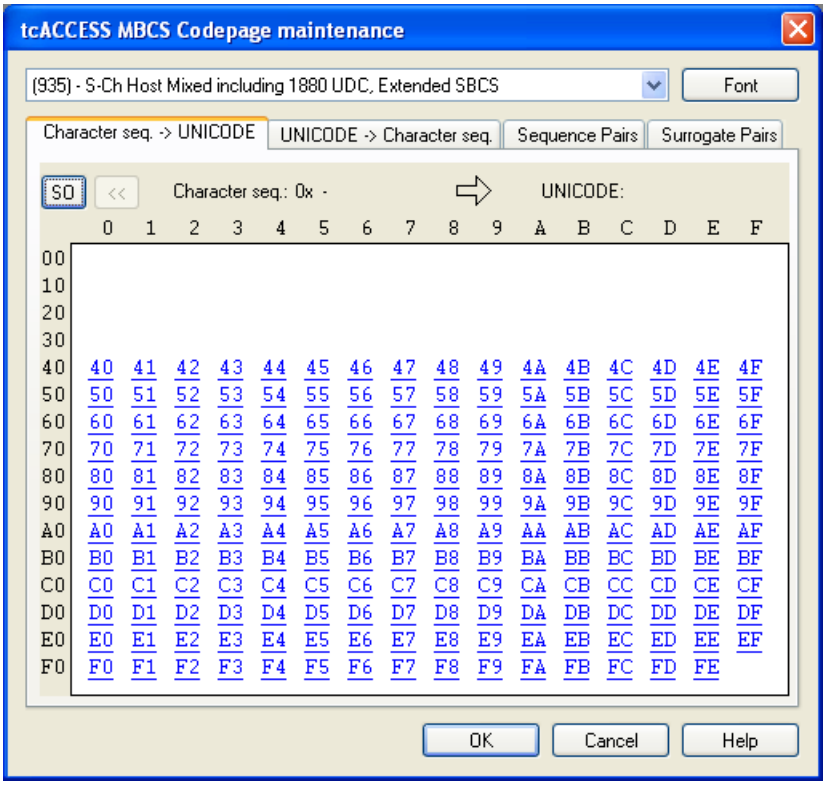


Fig. 144: Tab "Character seq. -> UNICODE" with "SO"-button

Button "Font"

This button displays a dialog where a font can be selected. The font is used for the alternate display of the converted UNICODE characters and should match the selected codepage.

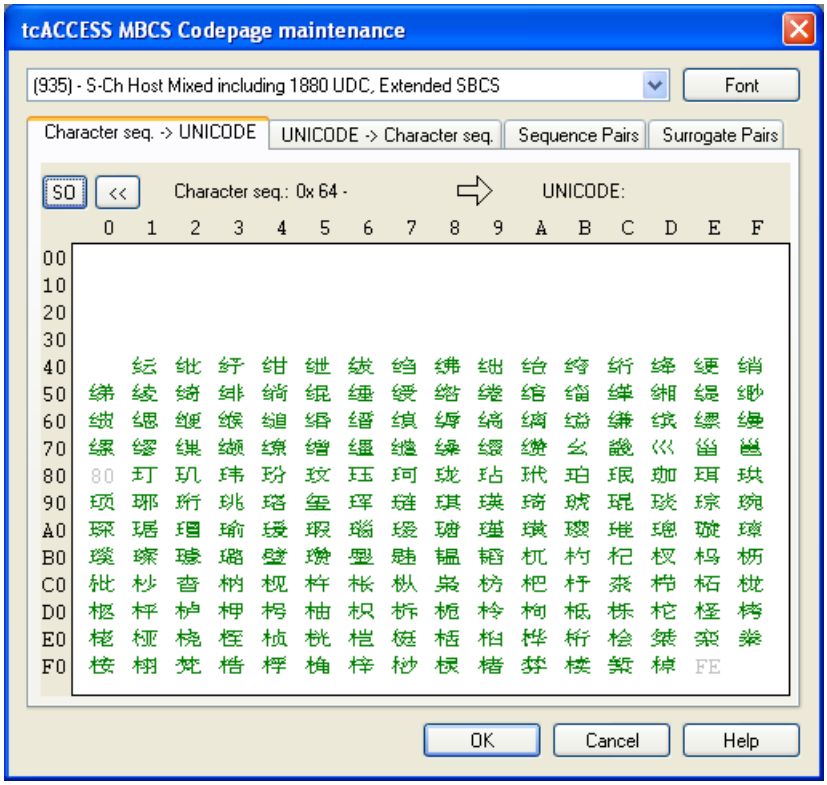


Fig. 145: Tab "Character seq. -> UNICODE" with alternate display

2.2.4.9.1.1 Change codetable entry

Use this dialog to maintain the "Character seq. -> UNICODE" conversion.

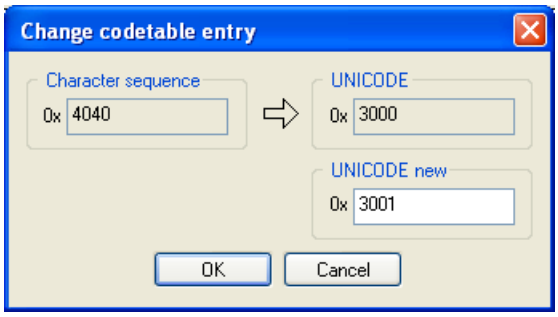


Fig. 146: Dialog "Change codetable entry" for "Charater seq. -> UNICODE"

The dialog components have the following meaning:

Character sequence

This field shows the character sequence that should be converted

UNICODE

This field displays the current UNICODE value.

UNICODE new

Specify the new UNICODE value that should be used for the conversion.

OK

Close the dialog and apply the changes.

Cancel

Close the dialog. Changes will not be applied.

2.2.4.9.2 Tab "UNICODE -> Character seq."

This tab shows all available conversions from UNICODE to character sequences. User defined codepages can also be edited.

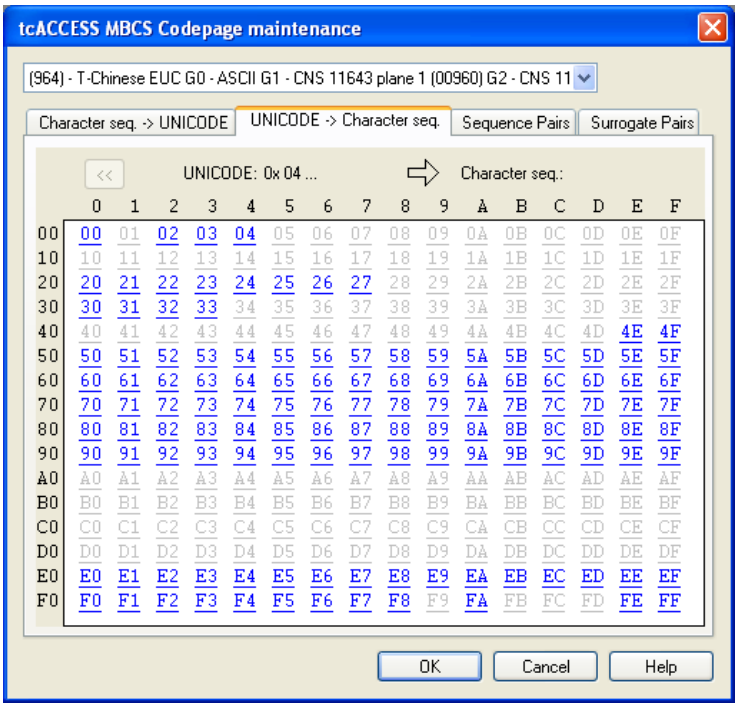


Fig. 147: Tab "UNICODE -> Character seq."

The dialog components have the following meaning:

Conversion table

This table shows all existing and possible conversions from a UNICODE value into the corresponding character sequence.

Moving the mouse pointer across the entries the corresponding conversions are being displayed in the upper part of the table.

A left mouse click on an entry of a user defined table can result in an action. Possible actions and the meaning of the table entries are:

blue entries: This value is only the first part of a UNICODE value (i.e. 0x04 in figure 147. There is at least one UNICODE value in this codepage that starts with this byte (there is a conversion for at least one of the UNICODE values 0x0400 to 0x04FF). A left mouse click displays the next level of the UNICODE value (that are all conversions of UNICODE values that consist of the combination with the (blue) value that has already been clicked on), hence a UNICODE value can be composed with values of two bytes.

Grey entries: This value is only the first part of a UNICODE value (i.e. 0x01 in figure 147). However this codepage does not contain a UNICODE value starting with this byte (hence no conversion for UNICODE values 0x0100 to 0x01FF). A left mouse click displays the next level of the UNICODE value (that are all conversions of UNICODE values that consist of the combination with the (blue) value that has already been clicked on), hence a UNICODE value can be composed with values of two bytes.

Green entries: A conversion of this UNICODE value into a character sequence exists (i.e. The conversion from 0x0451 to 0x8EADA3E7 in figure 149). A left mouse click displays a dialog that allows to change the conversion (refer to Change codetable entry, page 236).

Green entries: This UNICODE value is part of at least one UNICODE Sequence Pair, i.e. a composite UNICODE pair (as an example the following conversion exist: 0x0254 to 0xD890 and 0x0254 0300 to 0xECC3 in figure 149). A left mouse click changes to tab "Sequence Pairs" and the selected entry will be marked.

Grey entries: No conversion of this UNICODE value into a character sequence exists, however a conversion may be possible (example: UNICODE value 0x0400 in figure xxx has no character sequence assigned). A left mouse click display the dialog that allows the creation of a conversion (refer to Change codetable entry, page 236).

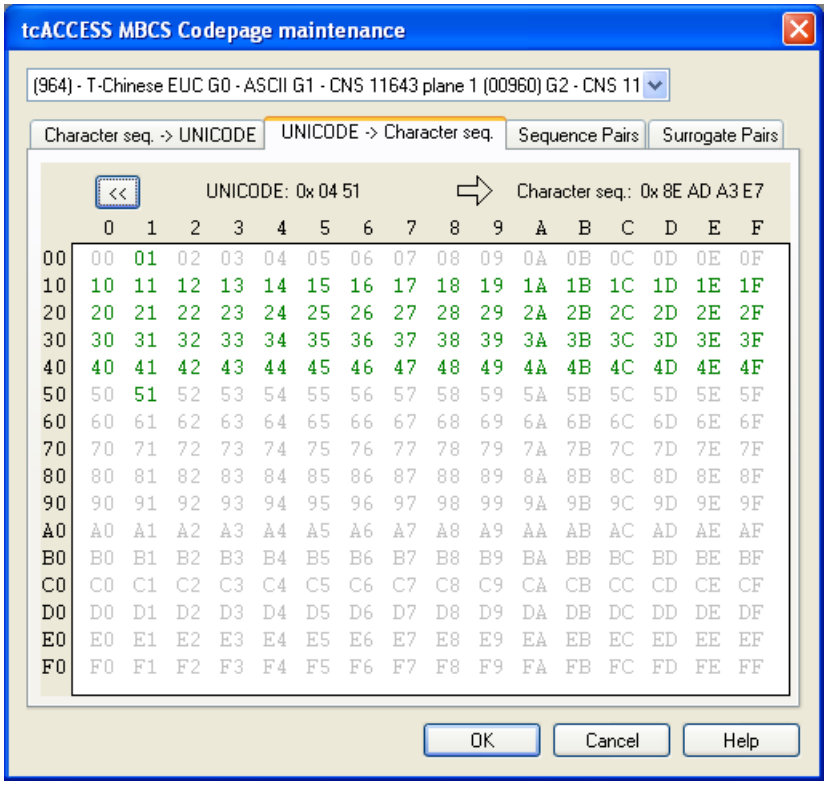


Fig. 148: Tab "UNICODE -> Character Seq."

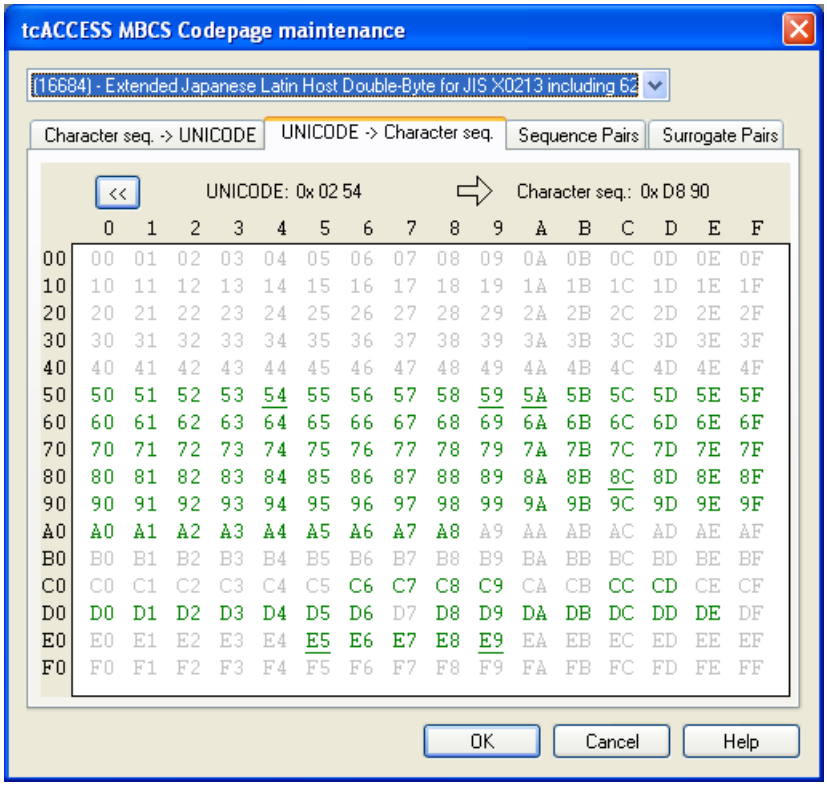


Fig. 149: Tab "UNICODE -> Character seq." with combined UNICODE Sequence Pairs

2.2.4.9.2.1 Change codetable entry

Use this dialog to maintain the conversions for "UNICODE -> Character seq."

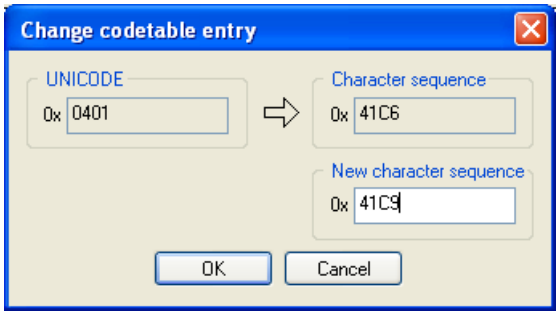


Fig. 150: Dialog "Change codetable entry" for "UNICODE -> Character seq."

The dialog components have the following meaning:

UNICODE

This field displays the UNICODE value.

Character sequence

This field displays the current value.

Character suquence new

This filed contians the new character sequence that should be used for the UNICODE value.

OK

Closes the dialog and applies the changes.

Cancel

Closes the dialog an no changes will be applied.

2.2.4.9.3 Tab "Sequence Pairs"

This tab displays the available conversions between character sequences and UNICODE Sequence Pairs. For user defined codepages the Sequence Pairs can also be edited. A Sequence Pair is a UNICODE pair that consists of 2 UNICODE values (i.e. 0x02540300).

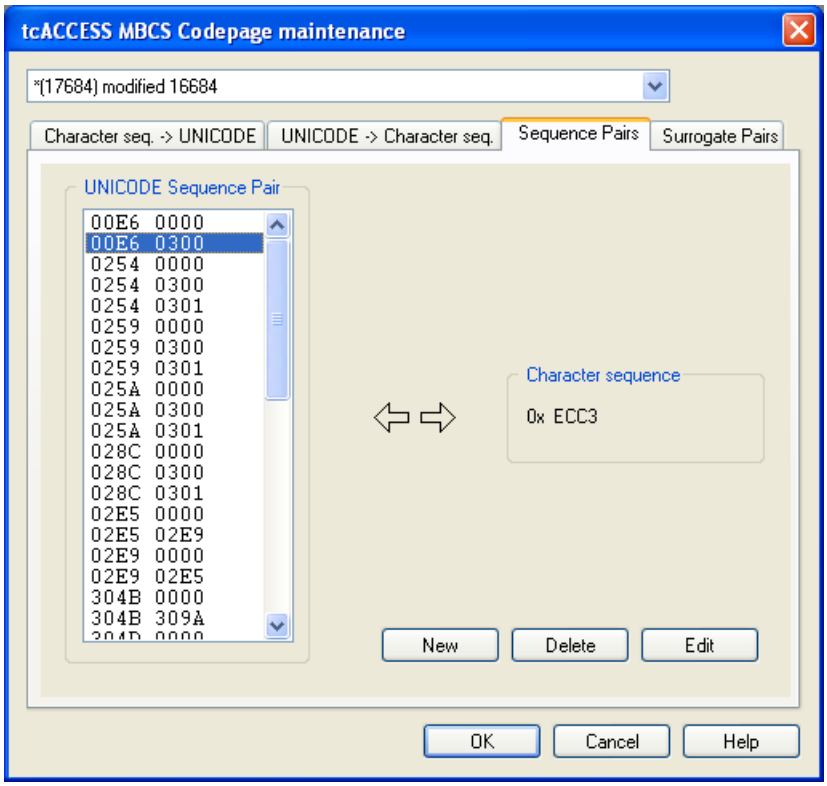


Fig. 151: Tab "Sequence Pairs"

The dialog components have the following meaning:

UNICODE Sequence Pair

The list displays the available UNICODE Sequence Pairs.

Character sequence

Displays the character sequence used to convert the selected UNICODE Sequence Pair from the list.

New

Displays a dialog to create new conversions (refer to New UNICODE Sequence Pair, page 238). The button is only active for user defined codepages.

Delete

This button deletes the conversion that has been selected from the list. The button is only active for a user defined codepage. If a Sequence Pair has been selected that ends with 000 (i.e. 0x02540000 in figure 148), the button is always inactive.

Edit

This button displays a dialog that allows the maintenance of a conversion (refer to New UNICODE Sequence Pair, page 238). The button is only active for user defined codepages.

2.2.4.9.3.1 New UNICODE Sequence Pair

Specify a conversion between a character sequence and a UNICODE Sequence Pair.

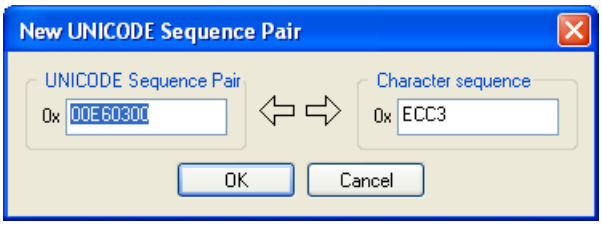


Fig. 152: Dialog "New UNICODE Sequence Pair"

The dialog components have the following meaning:

UNICODE Sequence Pair

Specify the UNICODE Sequence Pair for the conversion.

Character sequence

Specify the character sequence for the conversion.

OK

Close the dialog and apply the changes

Cancel

Closes the dialog but no changes will be applied.

2.2.4.9.3.2 Edit UNICODE Sequence Pairs

Use this dialog to edit an existing conversion between a UNICODE Sequence Pair and a character sequence.

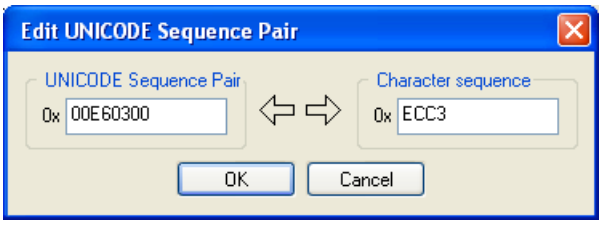


Fig. 153: Dialog "Edit UNICODE Sequence Pairs"

The dialog components have the following meaning:

UNICODE Sequence Pair

Edit the UNICODE Sequence Pair of the conversion.

Character sequence

Edit the character sequence of the conversion.

OK

Closes the dialog and applies the changes.

Cancel

Closes the dialog but no changes will be applied.

2.2.4.9.4 Tab "Surrogate Pairs"

This tab displays the available conversions between character sequences and UNICODE Surrogate Pairs. For user defined codepages they can be edited. A Surrogate Pair is a UNICODE value that consists of three bytes instead of two (i.e. 0x02000B).

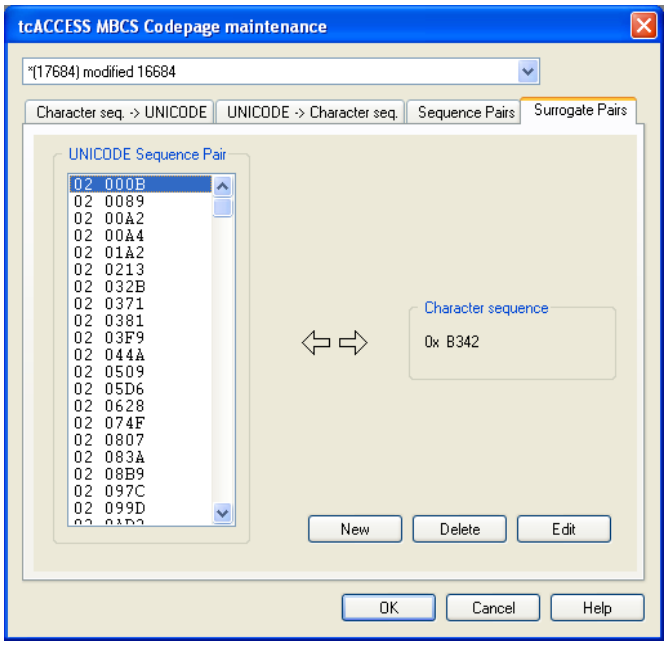


Fig. 154:Tab "Surrogate Pairs"

The dialog components have the following meaning:

UNICODE Surrogate Pair

The list displays the available Surrogate Pairs

Character sequence

The character sequence used to convert the UNICODE Surrogate Pair from the list (and vice versa).

New

Displays a dialog to create new conversions (refer to New UNICODE Surrogate Pair, page 241). The button is only active for user defined codepages.

Delete

This button deletes the conversion that has been selected from the list. The button is only active for a user defined codepage.

Edit

This button displays a dialog that allows the maintenance of a conversion (refer to Edit UNICODE Surrogate Pair, page 241). The button is only active for user defined codepages.

2.2.4.9.4.1 New UNICODE Surrogate Pair

Use this dialog to create a new conversion between a UNICODE Surrogate Pair and a character sequence.

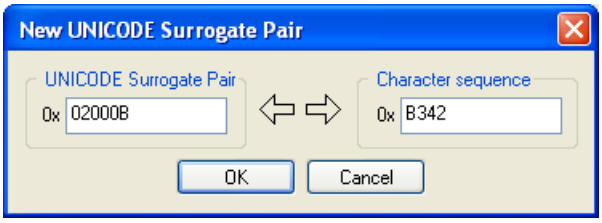


Fig. 155: Dialog "New UNICODE Surrogate Pair"

The dialog components have the following meaning:

UNICODE Surrogate Pair

Use this field to specify the Surrogate Pair for the conversion.

Character sequence

Use this field to specify the character sequence for the conversion.

OK

Closes the dialog and applies the changes

Cancel

Closes the dialog, but no changes will be applied.

2.2.4.9.4.2 Edit UNICODE Surrogate Pair

Use this dialog to edit an exsiting conversion between a UNICODE Surrogate Pair and a character sequence.

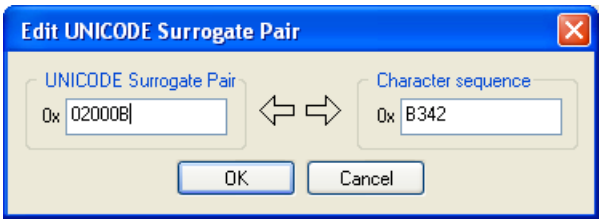


Fig. 156: Dialog "Edit UNICODE Surrogate Pair"

The dialog components have the following meaning:

UNICODE Surrogate Pair

Use this field to edit the UNICODE Surrogate Pair.

Character sequence

Use this field to edit the character sequence.

OK

Closes the dialog and applies the changes

Cancel

Closes the dialog but no changes will be applied.

2.2.4.9.5 New user defined table

When selecting entry "New user defined table" in the selection list of available MBCS codepagesa dialog is displayed where the properties of the new table can be specified.

Fig. 157: Dialog "New user defined table"

The dialog components have the following meaning:

CCSID of the new codetable

Use this field to specify a new CCSID for the new user defined table. Valid values are between 0 and 99998.

Description for new codetable

Maximum 20 characters can be used in this field.

Template for new codetable

Select an existing codetable from the list that should be used as a template for the new user defined table. Individual adaptations to the table can be applied ata later stage.

OK

Creates the user defined codetable and closes the dialog. If the CCSID is invalid or already exists an error message is issued and the dialog is not closed.

Cancel

Closes the dialog but no user defined table will be created.

2.2.4.10 SQL Analyzer...

A tcACCESS user who has signed on as Administrator (refer to Administrator , on page 186) can use this dialog to analyze SQL-statements and to determine the efficiency of queries that have been executed. This will increase the performance of applications. The tcACCESS SQL-Engine performs the analysis and provides an accurate performance profile.

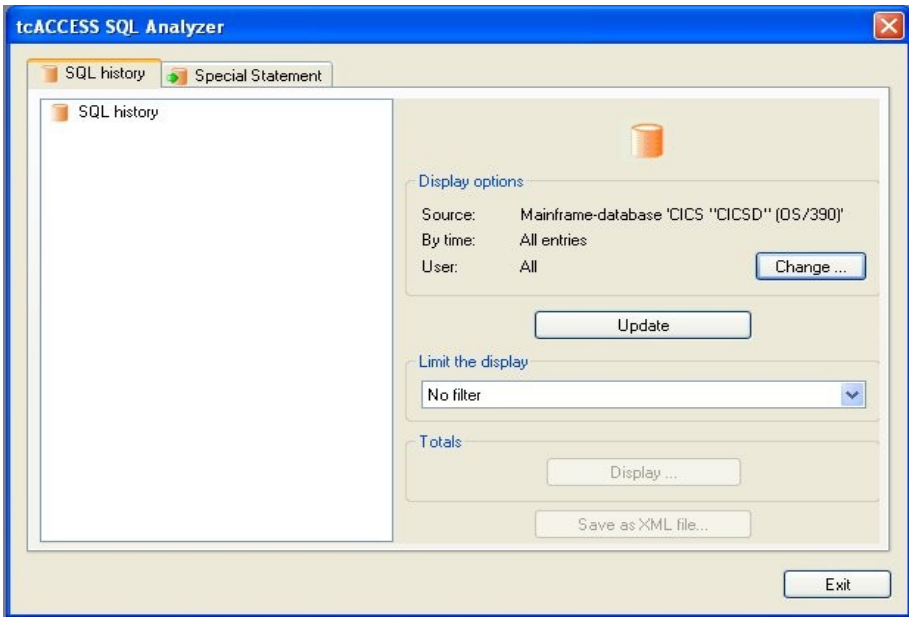


Fig. 158: Dialog „SQL Analyzer"

The dialog components have the following meaning:

SQL history

SQL statements that have been executed can be evaluated and their efficiency can be analyzed (refer to SQL-History, on page 244).

Special statement

Using this tab a special SQL statement can be analyzed (refer to Analyze special statement, on page 263).

Exit

Terminates the dialog.

2.2.4.10.1 SQL-History

The SQL-History can be used to analyze the efficiency of queries that have already be executed.

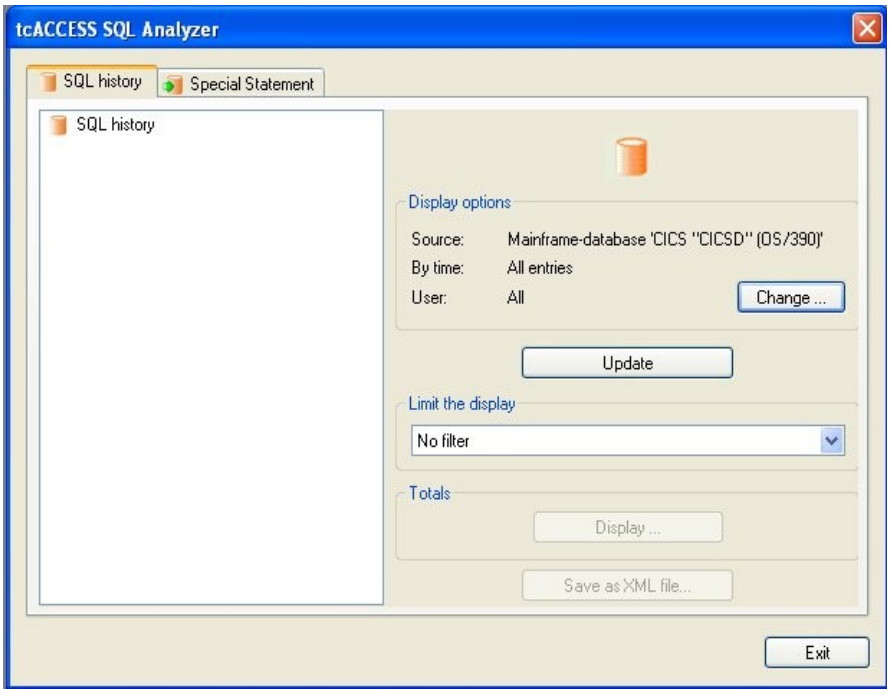


Fig. 159: Dialog „SQL-Historie"

The dialog components have the following meaning:

Tree view

Queries that have been executed and their analysis results are displayed in this tree view.

Display options

Options are displayed that can influence the SQL history. Button ”Change...” can be used to modify these options.

Change...

The display options can be changed. A dialog will be displayed (refer to Display options, on page 246).

Update

When this button is pressed the SQL history information are being retrieved using the specified display options.

Limit the display

Select one of the filters to limit the amount of information displayed in the tree view. The following filters can be used:

No filter:

All queries will be displayed

Statements with EC greater 80%:

All queries with an Efficient Coefficient greater than 80% will be displayed.

Statements with EC below 20%:

All queries with an Efficient Coefficient below 20% will be displayed.

Statements using full keys:

All queries with an Efficient Coefficient of 100% will be displayed.

Statements cancelled:

All statements with a return code of not equal to 0 will be displayed.

Statements reaching MAXIO:

All statements that reached the limit of MAXIO will be displayed.

Statements with long execution time:

All statements with an execution time greater than specified in the options will be displayed.

(Totals) Display...

Displays a statistic that is created from all retrieved statements (refer to Total statistic, on page 248). This button is only active when a SQL history has already been retrieved.

Save as XML file...

A SQL history that has been retrieved can be saved as an XML file. This may be useful when a history with a lot of entries has been created that should be evaluated and analyzed at a later time. Processing the XML file is faster than always retrieving the data from the host. The button is only active when a SQL history has already been retrieved.

2.2.4.10.1.1 Display options

This dialog can be used to define options that influence the retrieval of the SQL history.

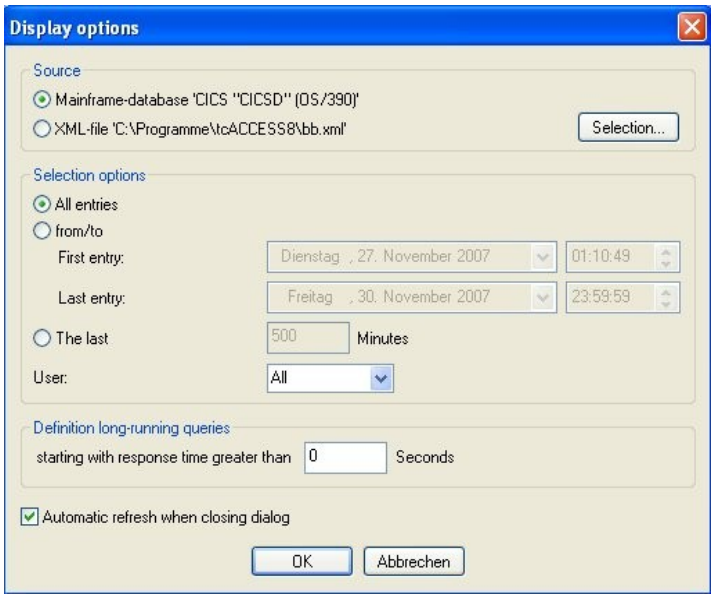


Fig. 160: Dialog „Display options"

The dialog components have the following meaning:

Source

This group can be used to specify where the data for the SQL history comes from. This can either be the host or an XML file. Select the corresponding check box. Option 'XML file' is only available when an XML file has been selected using button ”Selection”.

Selection...

This button displays a dialog where the XML file can be selected. The file must have been previously saved by the SQL Analyzer.

Selection Options

Using this group the selection of the SQL statements can be limited to time intervals or to certain users.

All entries

No limits will be specified all entries should be retrieved.

from/to

The first and the last entry can be specified. All statements that fall between the specified borders are analyzed. The date and time of the mainframe is used.

The last x minutes

All statements that have been executed within the last n minutes are retrieved and analyzed. Specify the number of minutes in this field.

User

All statements of the SQL history that have been executed by the specified user will be analyzed. Existing users can be selected from the list box or the user name can be specified in the text field.

Definition long-running queries ...

Use this text field to specify the value (in seconds) that should be used to identify long running queries (Filter 'Statements with long execution time').

Automatic refresh when closing the dialog

When this option is active the analysis of the SQL history will be automatically started when this dialog is closed via the OK button.

OK

All specifications will be applied and the dialog will be closed.

Cancel

The dialog wioll be closed and no action is taken.

2.2.4.10.1.2 Total statistic

This statistic is created from all statements retrieved from the SQL history. It is meant to provide an overview about the efficiency of the SQL-Engine queries over a longer period of time.

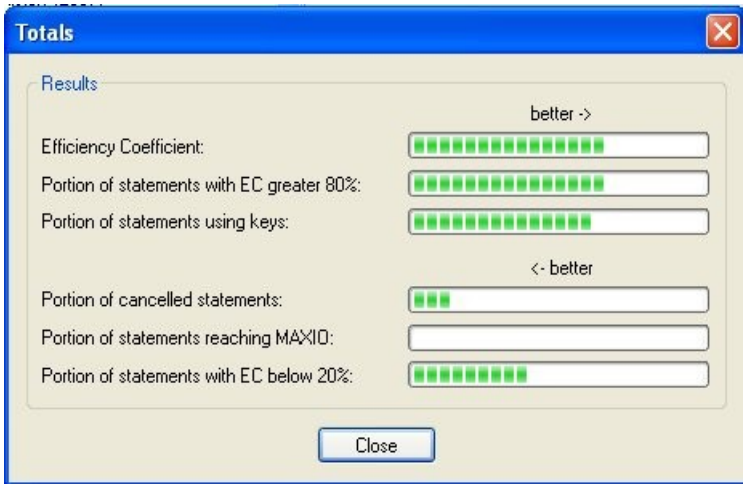


Fig. 161: Dialog „Totals"

The dialog components have the following meaning:

Efficiency coefficient

This value shows the arithmetic average of the efficiency coefficient of all processed queries. The better the value the more efficient are the queries.

Portion of statements with EC greater 80%

This value shows the portion of queries with an EC greater than 80% in relation to all processed queries.

Portion of statements using keys

This value shows the portion of queries with an EC of 100% in relation to all processed queries. These queries are of optimum qualitiy.

Portion of cancelled statements

This value shows the portion of cancelled statements in relation to all processed queries. A cancelled query is a query that did not return a return code of 0.

Portion of statements reaching MAXIO

This value shows the portion of queries in relation to all processed queries that have reached the MAXIO limit on the mainframe. Mostly those queries do not use a key. There might be potential for a better performance if these queries can be changed to use keys.

Portion of statements with EC below 20%

This value shows the portion of queries with an EC below 20% in relation to all processed queries.

Close

Close the dialog.

2.2.4.10.1.3 Result presentation

After the data has been retrieved, processed and analyzed the result is displayed in the tree view. Depending upon the entry selected in the tree view the display on the right changes. Next the dialog components for the individual tree view entries are discussed.

Entry „SQL-Command“:

Because the SQL history may contain multiple SQL commands, the commands are mapped with an execution timestamp and the name of the user who executed the statement.

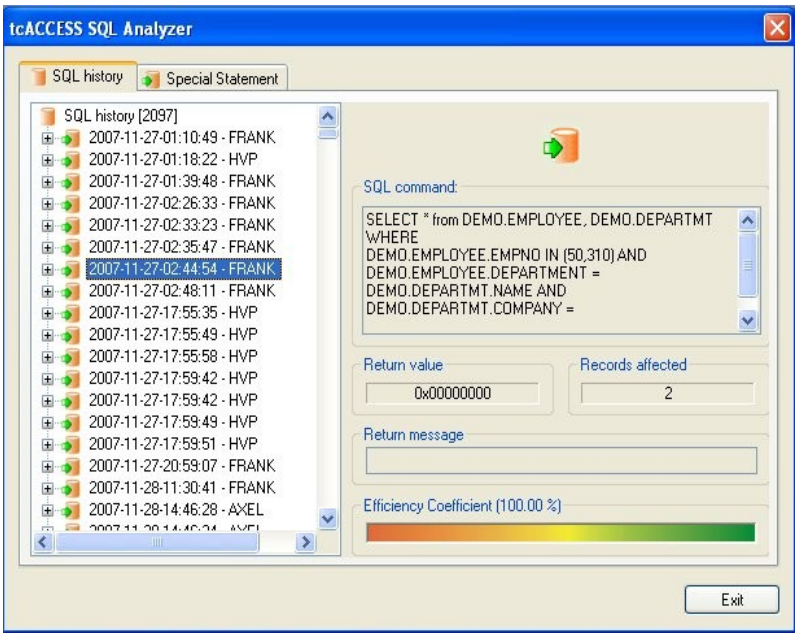


Fig. 162: History results ”SQL command“

The dialog components have the following meaning:

SQL command

Shows the statement that has been executed.

Return value

Shows the return code of the query.

Return message

Shows the message that has been returned and that corresponds to the return code.

Records affected

Shows the number of records that have been affected by this query. The value either represents the number of records returned (SELECT) or the number of records inserted, updated or deleted (INSERT/ UPDATE/ DELETE).

Efficiency coefficient

This display shows the efficiency of the SQL command. The further to the right the green bar is displayed the more efficient has been the statement.

Entry "Execution timesn":

Timing information are displayed for the SQL command.

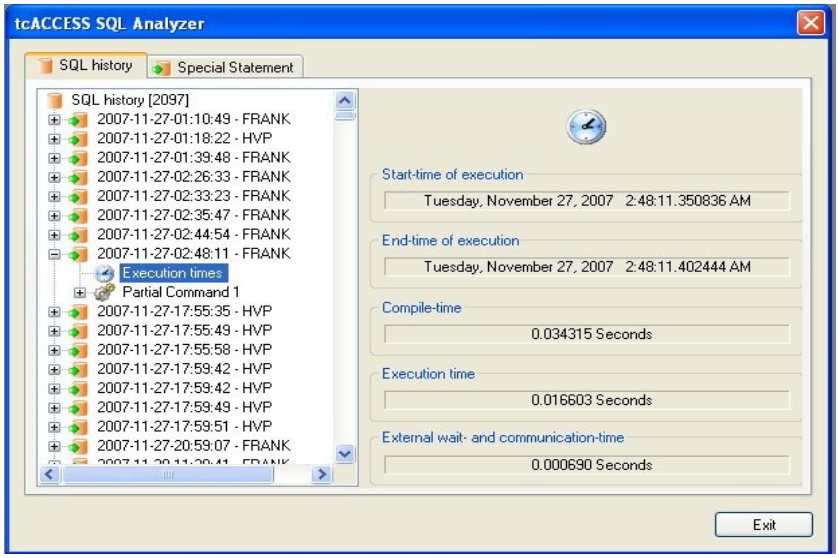


Fig. 163: Historiy results "Execution times"

The dialog components have the following meaning:

Start-time of execution

Time when the execution of the command started.

End-time of execution

Time when the execution of the command ended.

Compile-time

The time used by the host to compile the SQL command. The compilation process includes the break down of the statement into individual portions, the syntax check and the preparation of the actual data access.

Execution time

The time needed to execute the command and to read and process the data.

External wait- and communication-time

The external wait time used for the data-exchange between the host and the application. External wait times can happen when the application does not collect all records at once but may wait for a user response until the next block of data is collected..

Entry ”Partial command”:

A complex SQL statement can consist of multiple parts, hence multiple entries can exist that are sequence numbered.

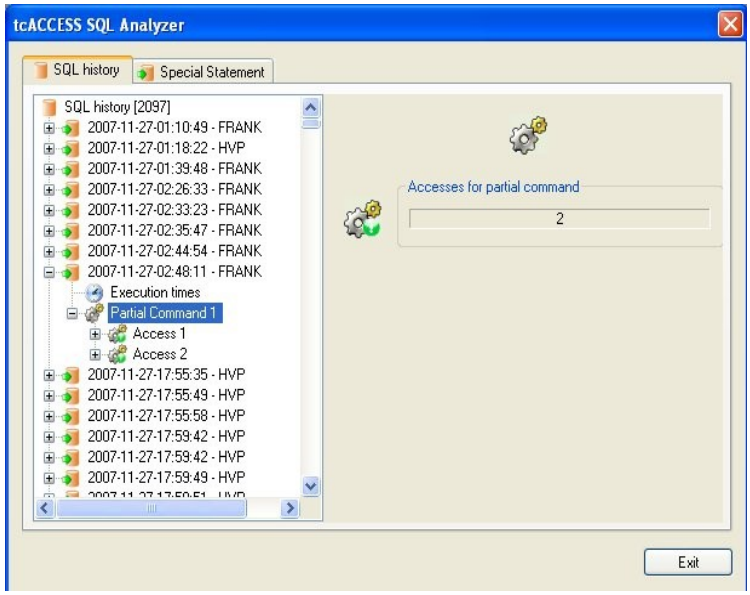


Fig. 164: History result ”Partial command”

The dialog components have the following meaning:

Accesses

The value represents the number of access required to execute the partial command. The colored coding of the symbol has the following meaning:


All accesses of the partial command have used an index.


Some but not all access of the partial command have used an index.


No access of the partial command has used an index.

Entry "Access":

Every partial command may consist of multiple accesses. Because of this the tree view may contain multiple entries which are sequence numbered.

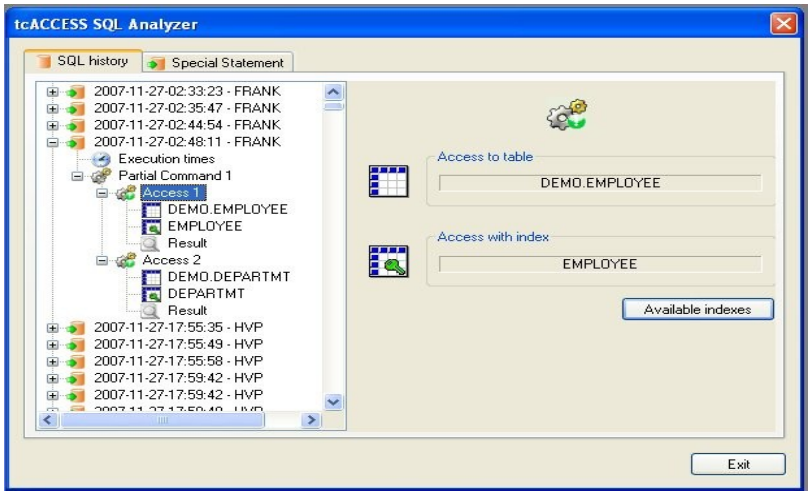


Fig. 165: History results "Access"

The dialog components have the following meaning:

Access to table

Name of the table that has been accessed during the execution of the partial command.

Access with index

Name of the index that has been used for the access. The index is used when the statement has used a field for a qualified WHERE condition that is an index of the file. If no index has been used the following test is displayed "Access with no index".

Available indexes

This button displays all available indexes for this table in the tree view. Use this information to make the query more efficient.

Entry ”Table”:

The table used for the access is displayed.

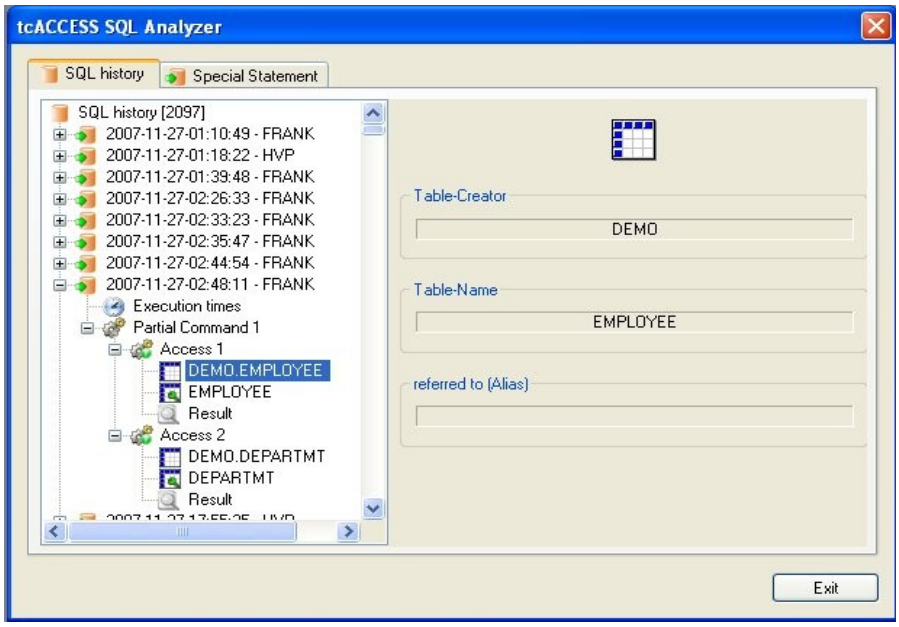


Fig. 166: History result ”Table”

The dialog components have the following meaning:

Table-Creator

Shows the name of the table creator.

Table-Name

Shows the name of the table

referred to (Alias)

Shows the alias name of the table.

Entry ”Index”:

Displays the index used for the access.

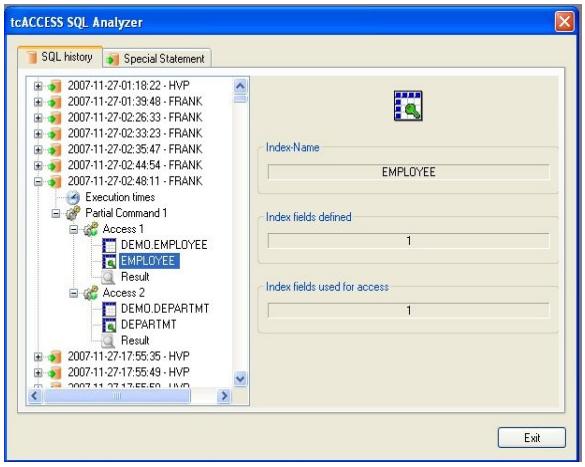


Fig. 167: History result ”Index”

The dialog components have the following meaning:

Index-Name

Displays the name of the index.

Index fields defined

Displays the numnber of index fields.

Index fields used for access

Displays the number of indexes used for the access.

The query can be initiated using a start value

This text is only displayed when a start value is available for an index field. It is not important that the full key has been supplied but the leftmost positions if the key must be supplied.

Note: This text is only displayed for tab ”Special statement”. It is not displayed for tab ”SQL history”.

Entry ”Available indexes”:

If button ”Available indexes” has been pressed for entry ”Access”, a new entry will be added to the tree view. If no indexes are defined for the table, the text for this entry is ”No indexes available”.

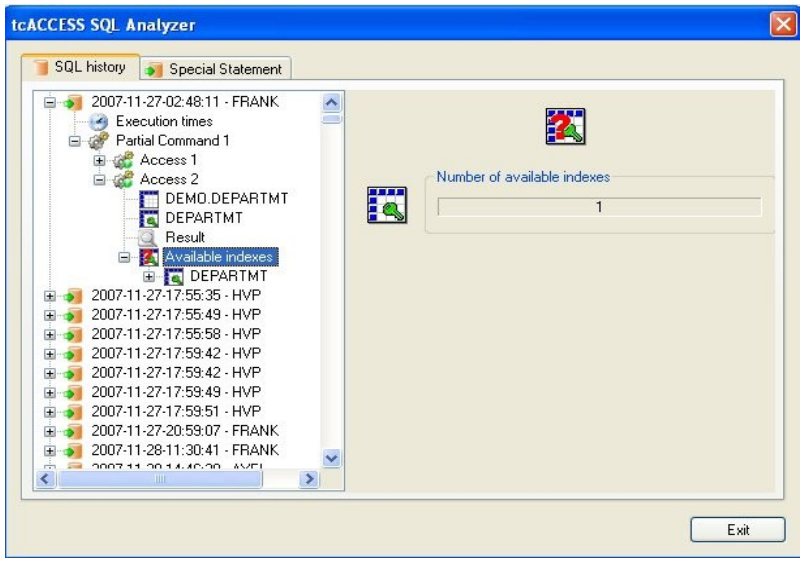


Fig. 168: SQL history "Available indexes"

The dialog components have the following meaning.:

Number of available indexes

Shows the number of index fields for this table.

Entry „Available index":

If indexes are available, they are displayed as new entries for tree view "Available indexes".

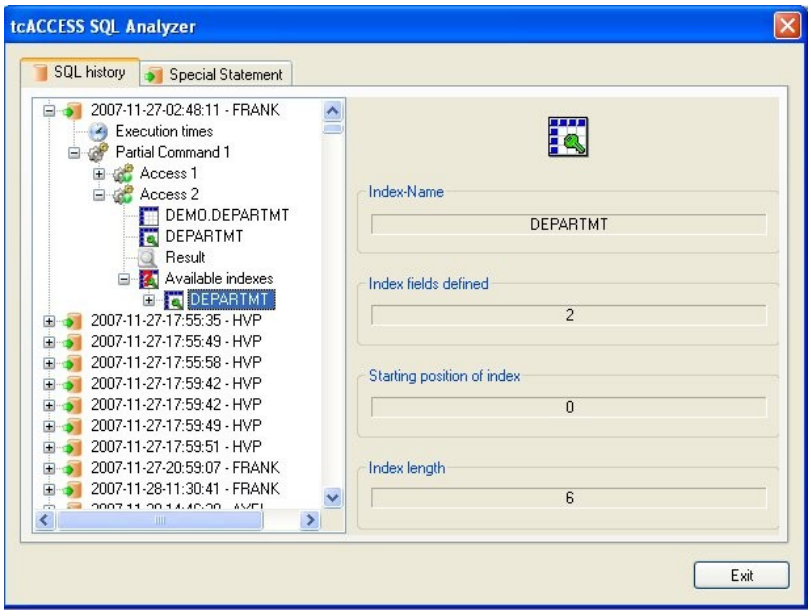


Fig. 169: SQL history "Available index"

The dialog components have the following meaning:

Index-Name

Shows the name of the index.

Index defined fields

Shows the number of index fields.

Starting position of index

Shows the position of the first byte of the index in a data record.

Index length

Shows the length of the index in bytes.

Entry "Index field":

For every available index the fields of the index become new entries in the tree view.

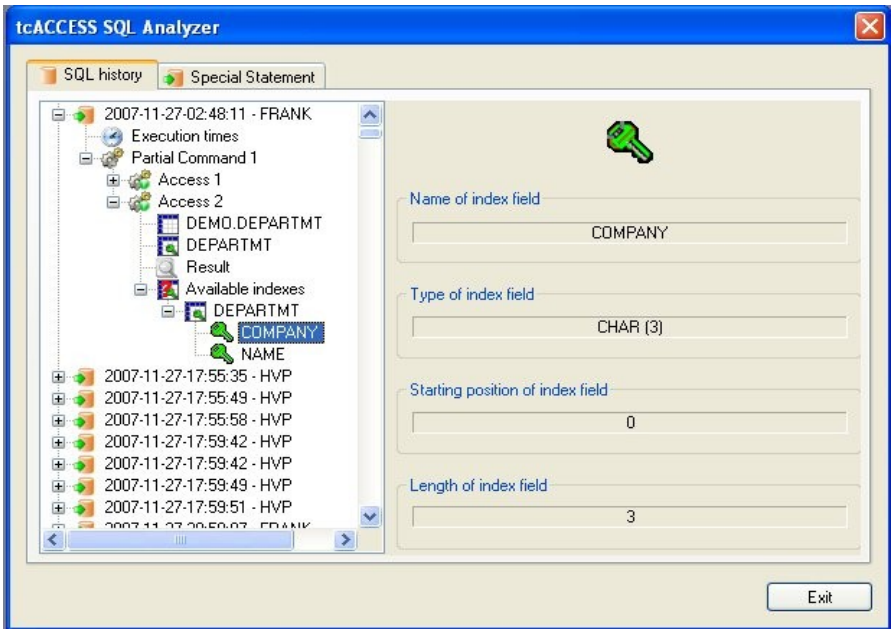


Fig. 170: SQL history „Index field"

The dialog components have the following meaning:

Name of index field

Shows the name of the index field.

Type of index field

Shows the type of the index field.

Starting position of index field

Shows the position of the first byte of the index field in a data record.

Length of index field

Shows the length in bytes of the index field.

Entry ”Result”:

The results are being displayed. They are related to an access to a certain table of the partial command.

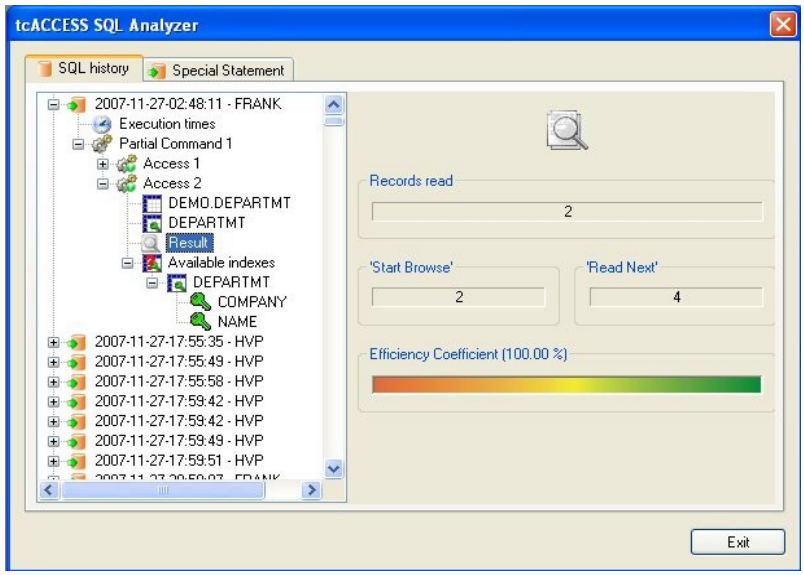


Fig. 171: SQL history ”Results”

The dialog components have the following meaning:

Records read

Shows the number of records read from the accessed table. If the access was not of type ”Read” (SELECT), the header will be ”Inserted records (INSERT)” or ”Updated records (UPDATE)” or ”Deleted records (DELETE)”.

'Start Browse'

Shows the number of Start-Browse operations performed for this access.

'Read Next'

Shows the number of Read-Next-operations for this access. The number is always one larger than the resultset to determine the change of the key. This is not the case if the number of records to be returned has been limited and the limit value has been reached.

Efficiency coefficient

This display hows the efficiency of the access. The further the green bar is to the right the more efficient has been the access of the partial command.

Entry ”Tables used:

This section gives an overview about all tables used by the analyzed statements.

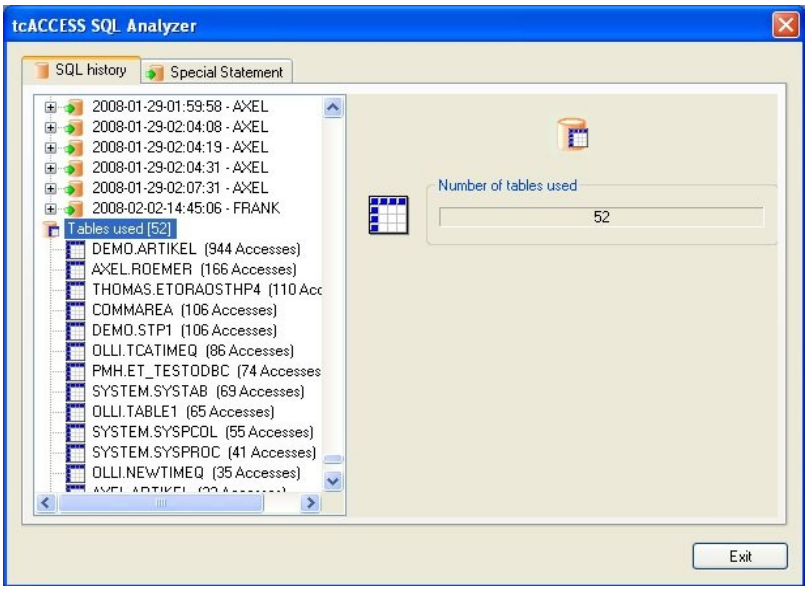


Fig. 172: SQL history ”Tables used”

The dialog components have the following meaning:

Number of tables used

Shows the number of tables used during the anlysis period.

Entry ”Table used”:

Display of statistical data for a selected table.

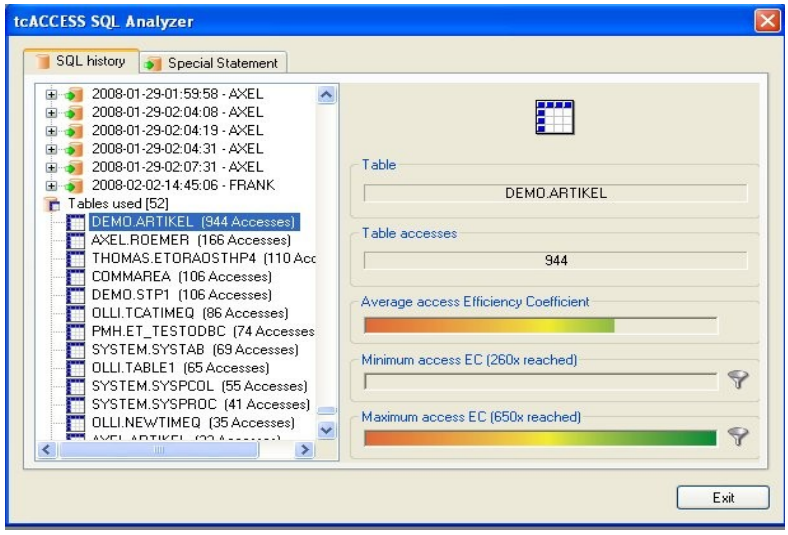


Fig. 173: SQL history ”Table used”

The dialog compüonents have the following meaning:

Table

Shows the name of the used table.

Table accesses

Shows the number of total accesses to this table by all analyzed statements

Average access Efficiency Coefficient

Shows the efficiency of the accesses to this table. The further the green bar is to the right the more efficient has been the access to this table.

Minimum access EC

Shows the worst EC that has been calculated. The number in brackets indicates how often an access has resulted in the EC. If the symbol at the right is pressed these statements will be filtered for the tree view.

Maximum access EC

Shows the best EC that has been calculated. The number in brackets indicates how often an access has resulted in the EC. If the symbol at the right is pressed these statements will be filtered for the tree view.

Button 

Using this button a filter is being set for the tree view. This filter only shows the entries that are related to the selected table and matches the criteria of the corresponding EC (min/max).

2.2.4.10.1.4 Increase efficiency

This chapter discusses how the SQL Analyzer can be used to increase the efficiency of SQL queries.

The example uses tables „EMPLOYEE" and „DEPARTMT", which consist of the following fields and indexes:

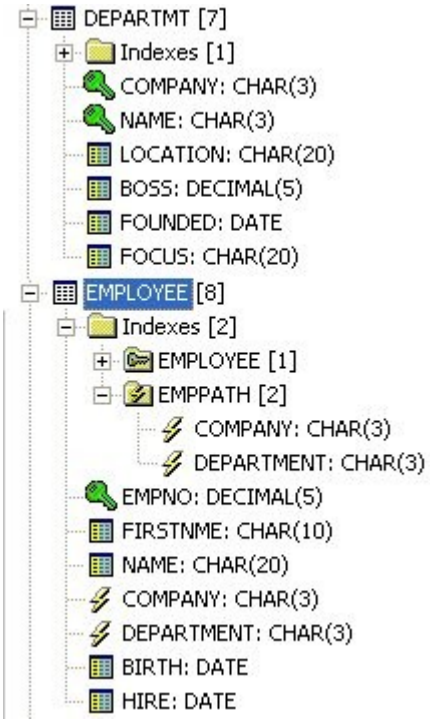


Fig. 174: Structure of sample tables

The analysis performed by the SQL Analyzer shows a bad EC for a query.

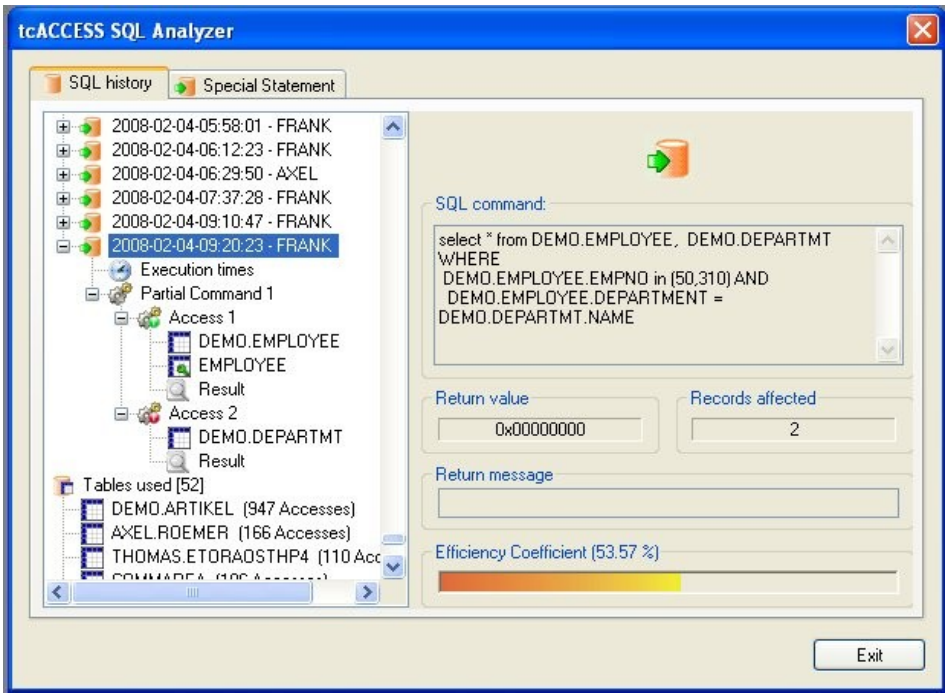


Fig. 175: Query showing a bad Efficiency Coefficient

Having a closer look at the individual accesses one can see that the second access has a high number of accesses but returns view records resulting in a bad EC.

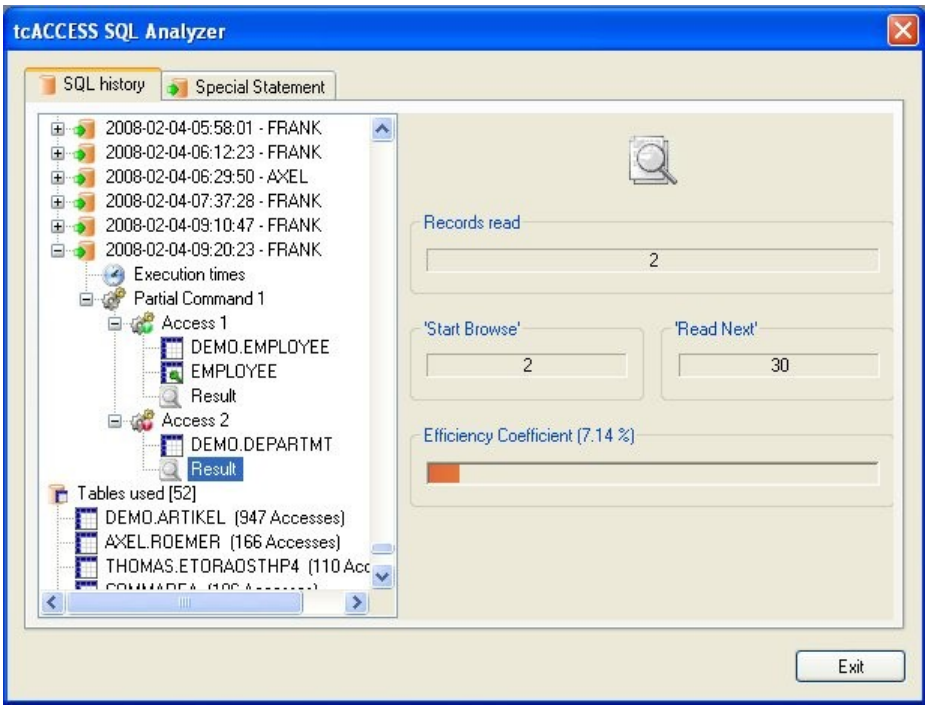


Fig. 176: Access with bad Efficiency Coefficient

The tree view to the right indicates that no index has been used for the access although the statement limited the search ”AND DEMO.EMPLOYEE.DEPARTMENT = DEMO.DEPARTMT.NAME“.

When looking at the available indexes for this access it can be seen that field Feld DEPARTMT.NAME is part of the index, but the index cannot be used in that way, because NAME starts at position 3 of the index.

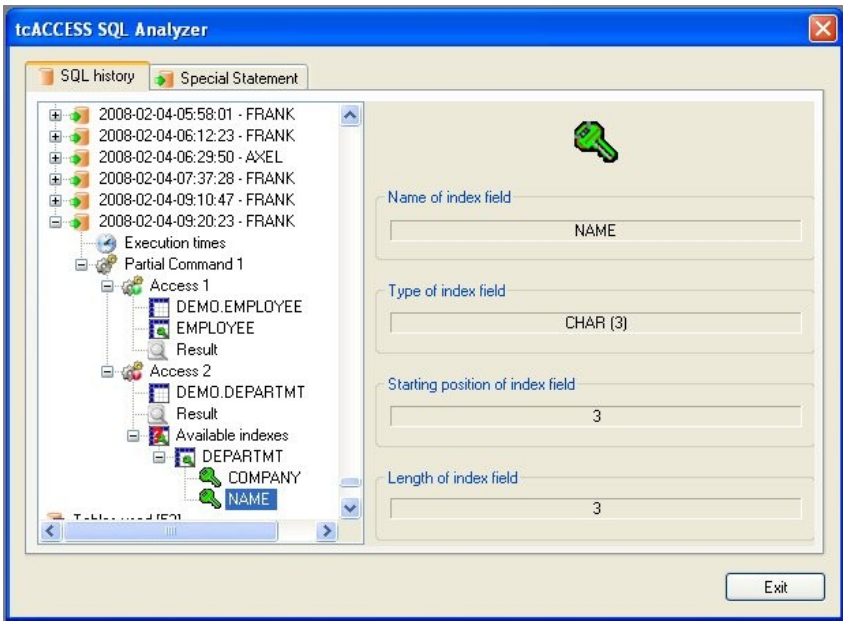


Fig. 177: Field of the available index

Next we change the SQL statement to also use the first part of the index and the EC really improves!

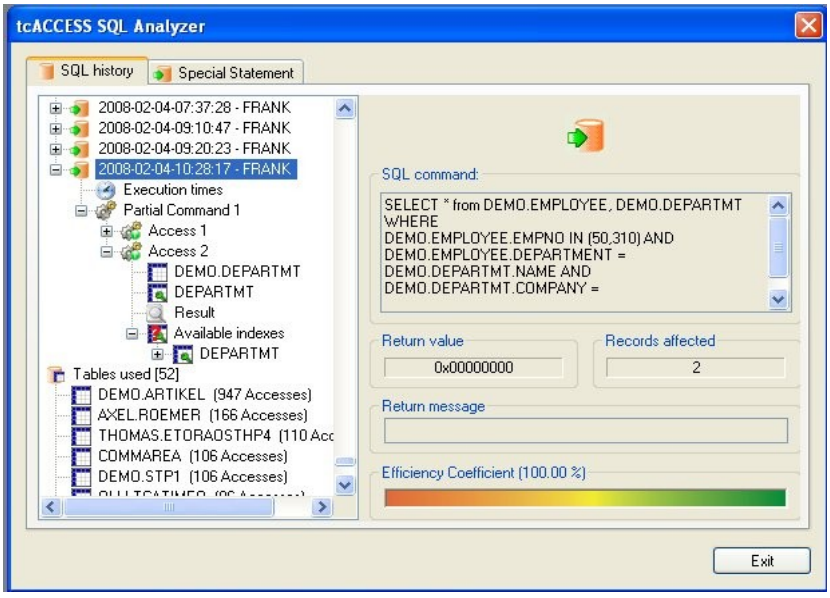


Fig. 178: Query with a good EC

Now the full index has been used to access the table,

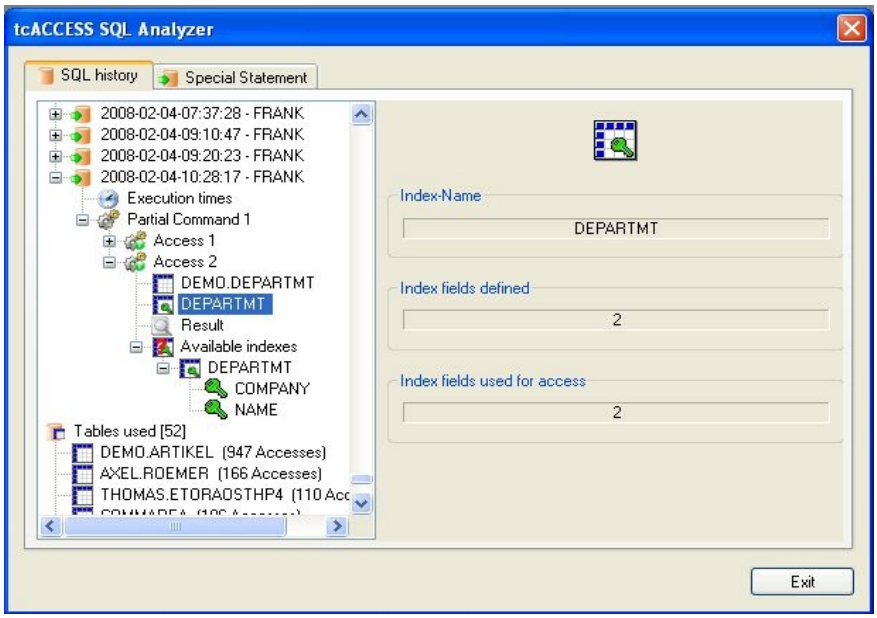


Fig. 179: Usage of the index

hence the number ”Read-Next” operations has been drastically reduced.

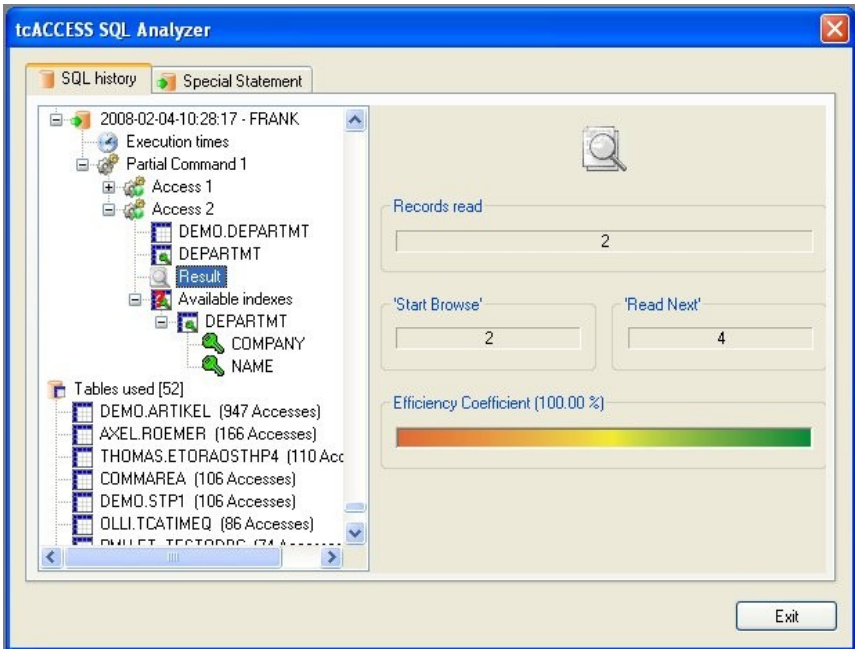


Fig. 180: Access with a good EC

2.2.4.10.2 Analyze special statement

During the analysis of a special statement the statement is checked concerning the tables and indexes that will be used when the statement is actually executed.

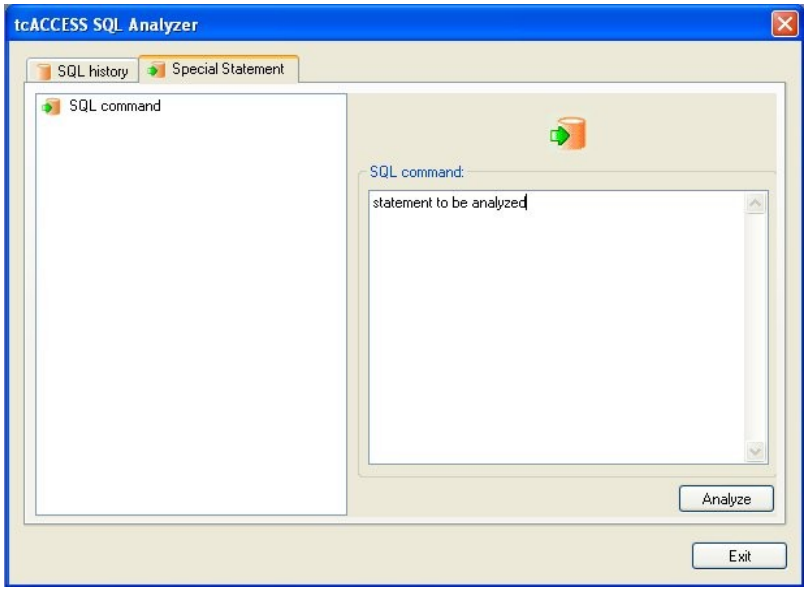


Fig. 181: Dialog "Analysis of a special statement"

The dialog components have the following meaning:

Tree view

When the analysis has been completed the results are displayed in the tree view.

SQL command

Use this field to specifiy the SQL statement that should be analyzed.

Analyze

Start the analyze process.

2.2.4.10.2.1 Presentation of the results

After the analysis has been completetd the results are displayed in the tree view. Depending upon the entry selected in the tree view the display on the right changes. In the following chapter the dialog components for the individual entries are being discussed.

Entry ”SQL command“:

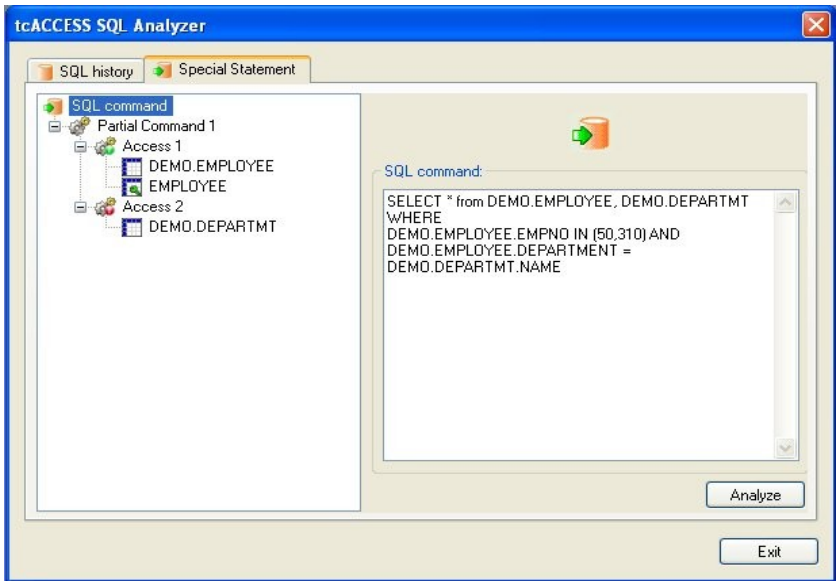


Fig. 182: Result ”SQL command

The dialog components have the following meaning:

SQL command

Shows the statejment that has been analyzed. For another analysis specify a new statement.

Analyze

This button starts the analysis process.

Remaining tree view entries:

All other tree view entries and their corresponding dialogs are identical to the entries in the SQL history and the corresponding dialogs (refer to Result presentation, on page 249).

2.2.5 The Menu "Window"

Several dialogs, such as FIELD DEFINITION IMPORT, EDIT SQL-ENGINE TABLE DEFINITION, PC => HOST TRANSFER, are displayed within the main program and can be accessed parallel (i.e., they must not be closed before further operations with the main menu can be performed). The WINDOW menu allows you to maintain these special dialogs.

2.2.5.1 Next

Switch to the next open dialog.

2.2.5.2 Previous

Switch to the previously open dialog.

2.2.5.3 Close

Close the current dialog and use another open dialog.

2.2.5.4 Cascade

Arrange all open dialogs in an overlapping manner starting from the upper left corner to the right bottom

2.2.5.5 Arrange

Arranged the open dialogs in a manner that allows you to use the free space within the tcACCESS main program completely.

2.2.5.6 Arrange icons

Arranged all minimized dialogs shown as symbols in a sequence.

2.2.5.7 Close all

Closes all open dialogs.

2.2.6 The Menu "Help"

The sub-menu HELP on the main menu contains all functions for displaying product information.

2.2.6.1 About tcACCESS

Contains information about tcACCESS, including, Product information, PC licence, Host licence and File versions.

2.2.6.1.1 Tab Dialog "Product information"

This part of the dialog contains the product name, the version number, the installed service pack with the highest number, and an Internet link to the tcACCESS home page. This allows you to download service packs without the need to enter the license codes.

Fig. 183: About tcACCESS - Tab Dialog "Product information"

2.2.6.1.2 Tab Dialog "PC licence"

This tab dialog displays the license codes for the PC components of tcACCESS. Also, new license information can be specified.

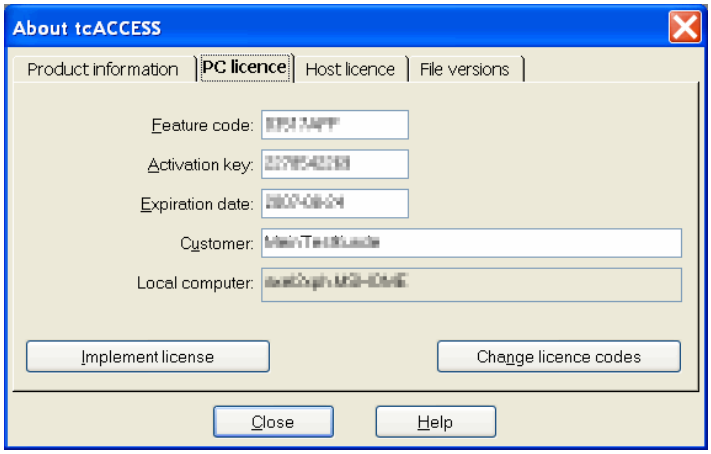


Fig. 184: About tcACCESS - Tab Dialog "PC licence"

The dialog components are described below:

Change license codes

If new license codes for the PC component are available, they can be defined in this screen and can be applied using this button. When the new license code has been successfully applied an informational window will be displayed confirming the action. The the license code is not correct the last license code will be active.

Implement license

A new dialog displays where the file can be selected that contains the license information. The next step is the selection of the license that should be implemented. Click "OK" to implement.

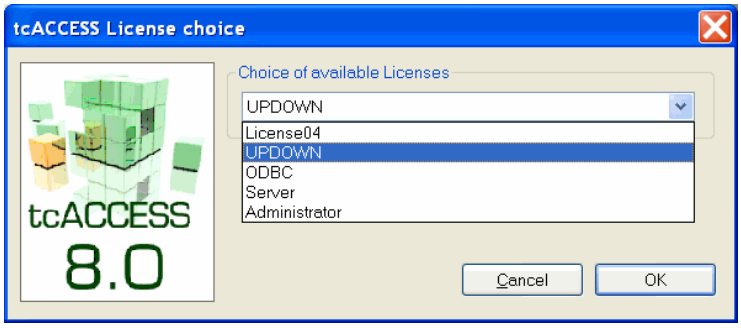


Fig. 185: About tcACCESS - Tab Dialog "PC licence" dialog "License choice"

When the new license code has been successfully applied an informational window will be displayed confirming the action.

2.2.6.2 Tab Dialog "Host licence"

This tab dialog displays the license codes for the Host components of tcACCESS.

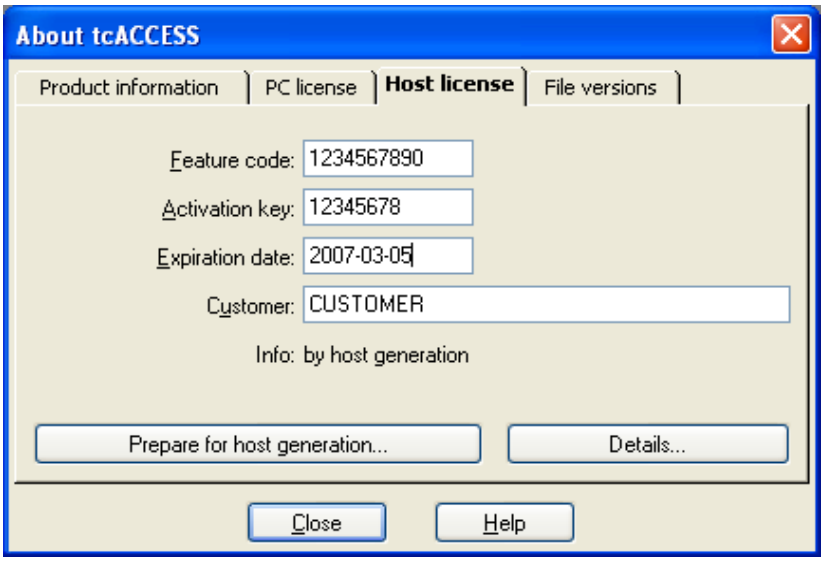


Fig. 186: About tcACCESS - Tab Dialog "Host licence"

The dialog components are described below:

License-Information

The four fields "Feature Code", "Activation key", "Expiration date" and "Customer" represent the basic license information. These information will be determined on the host, if a connection to the mainframe exists. If no connection exists, the values from the PC installation CD will be displayed.

Prepare for Host-Generation

The host license information from the tcACCESS installation CD must be used for the host tcACCESS generation. By pressing this button the relevant information will be displayed.

Details

This button is only available if a connection to the mainframe exists. By pressing this button the actually used license information on the host will be displayed.

2.2.6.3 Tab Dialog "File versions"

This tab dialog lists all modules that are in the tcACCESS installation directory. Their version number is also shown. Additionally, the highest module version number displays separately.

The following file types may appear in the display:

- Executable files (extension .EXE)
- Libraries (extension .DLL)
- PC Communication Handler Modules (extension .DVC)

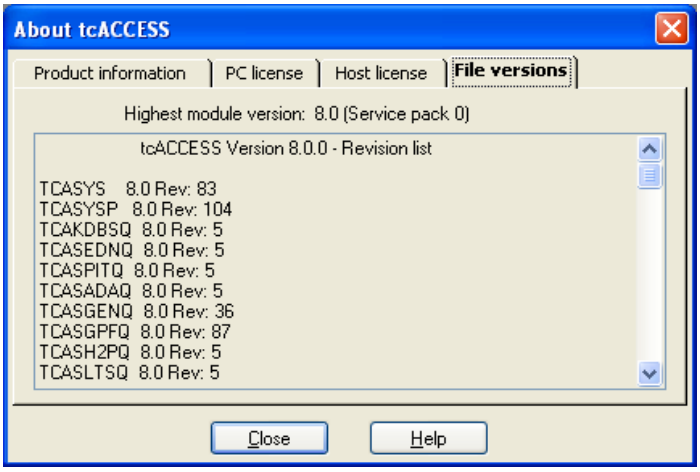


Fig. 187: About tcACCESS - Tab Dialog "File versions"

2.3 The Data Transfer Status Window

A window with status information is displayed each time a PC => Host Transfer (page *97*), Host => PC Transfer (page *127*) or Host SQL Query (page *156*) function is processed to exchange data between a PC and a mainframe.

It always displays in a standardized form, which is described in the following section.

2.3.1 Standard Display

The standard display shows an overview on the current status of a data transfer.

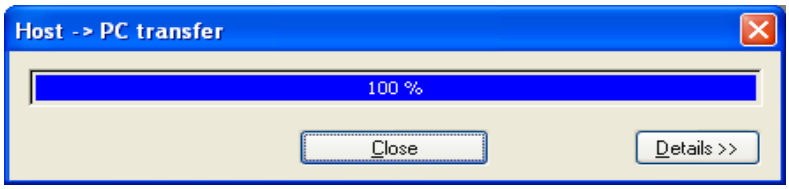


Fig. 188: Status Window for Data Transfer - Standard Display

The dialog components are described below:

Percentage display

The percentage display on the progress of a processed data transfer occupies the major part of the standard display. If the 100 % mark is reached, the **Cancel** button changes to **Close**.

For a PC => Host Transfer (page *97*) status window, tcACCESS is able to calculate the percentage of the transferred data during a transfer operation. In combination with Host => PC Transfer (page *127)* and Host SQL Query (page 156) functions, for performance reasons the calculation percentage does not perform unless the PROGRESS DISPLAY FOR DATA TRANSFER FROM HOST of the Options dialog (see page *93)* is active.

Cancel / Close

If the **Close status window automatically** button is set in the Options (see page *93*) dialog, it is not necessary to click the **Close** button after a completed data transmission. If you prefer to analyze the information provided under **Details**, it is recommended that you deactivate this button.

Details

Selecting this button displays or hides the extended information on a currently processed data transfer (see Details Display).

2.3.2 Details Display

This status window displays after selecting the **Details** button. It is identical to the upper section of the Standard Display while the extended information displays in the lower part.

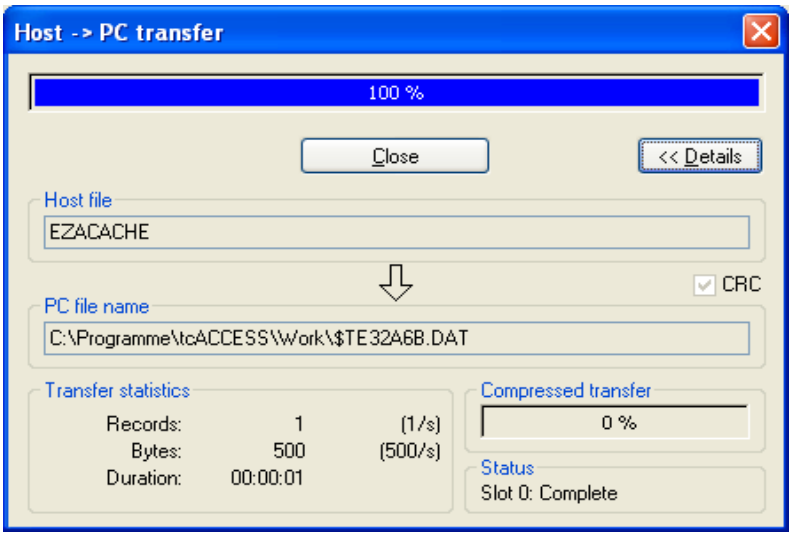


Fig. 189: Status Window for Data Transfer - Details Display

The dialog components are described below:

Percentage display, Cancel / Close, Details
See Standard Display*, page 270* for an explanation of this component.

Data source
This field displays the source of the data transfer. It can be either a PC file name, Table, Host file, or SQL command.

Data target
This field specifies the target for the data. It can be a PC file name, Table, or Host file.

Records sent
Shows the number of transmitted records. The value should always be identical with the value shown in the RECORDS RECEIVED field.

Records received
Shows the number of records received. The value should always be identical with the value shown in the RECORDS SENT field.

CRC
Indicates if CRC checksums have been used for the transmission.

Transfer statistics
This field displays some information about the current transfer. The number of sent records and bytes is shown here. Next to it, the transfer rate per second is displayed. The duration of the data transfer also displays.

Compressed transfer
If the **Compressed data transfer** button of the Options dialog (see page *93*) is active, this bar and the percentage shows the compression rate achieved for this data transfer operation. The higher the value, the better the compression results.

Status
Displays the current status of tcACCESS connection slot. Possible values are: Slot n:

MENU ITEMS	DESCRIPTION
Send	Data is sent to the host.
Receive	Data is received from the host.
Save	Writing a file on the PC.
Wait	Waiting for host response.
Aborted	Transfer cancelled.
Complete	Transfer successfully completed.

2.4

2.5 The Command Line Parameters

Command line parameters allow you to process many of the tcACCESS program functions automatically. The following parameters are supported:

/SL 123

Specify connection slot

123 => Slot number (between 0 and 254)

This parameter specifies the number of the connection slot. This parameter has a higher priority than "/FSF" and "/FSS".

If no parameter specified to define a slot ("/SL", "/FSF" oe "/FSS"), tcACCESS checks environment variable "DEFAULT_SLOT". This variable always exists, when a tcACCESS program has been started using the "tcACCESS Command Shell" (refer to The tcACCESS Command Shell BMLSHELL page *315)* and the variable contains a slot number. If this slot number does not exist, the slot number most recently used by the tcACCESS program for data communication between the PC and the mainframe will be used.

/FSF xyz

Find a slot, which uses the defined configuration

Requirement: There must be no host session active ("Find Slot Free")

xyz => Configuration name

tcACCESS tries to find a free slot which uses the defined configuration and has no active host session. If parameter "/SL" has been defined, it has a higher priority and prohibits the usage of "/FSF".

If no parameter specified to define a slot ("/SL", "/FSF" oe "/FSS"), tcACCESS checks environment variable "DEFAULT_SLOT". This variable always exists, when a tcACCESS program has been started using the "tcACCESS Command Shell" (refer to The tcACCESS Command Shell BMLSHELL page *315)* and the variable contains a slot number. If this slot number does not exist, the slot number most recently used by the tcACCESS program for data communication between the PC and the mainframe will be used.

/FSS xyz

Find a slot, which uses the defined configuration

<u>Requirement:</u> There must be an allocated session which is currently not in use
("Find Slot with Session")

xyz => Configuration name

tcACCESS tries to find a free slot which uses the defined configuration and has an allocated session. The session must not be active at the moment (e.g. active data transmission).

If all slots which might fit do have an active session, you can use the parameter "/ON". TcACCESS then tries to find a corresponding slot with no session and tries to establish a new connection.

In combination with option "Close inactive connection after.." /FSS can be used to do a "Connection Pooling" (refer to page 205).

If "/SL" has been specified, this parameter has a higher priority and prohibits the useag of "/FSS".

If no parameter specified to define a slot ("/SL", "/FSF" oe "/FSS"), tcACCESS checks environment variable "DEFAULT_SLOT". This variable always exists, when a tcACCESS program has been started using the "tcACCESS Command Shell" (refer to The tcACCESS Command Shell BMLSHELL page *315)* and the variable contains a slot number. If this slot number does not exist, the slot number most recently used by the tcACCESS program for data communication between the PC and the mainframe will be used.

/ON

Starting a session
This parameter allows you to start a mainframe session. It is not necessary that a host connection already exist (parameter '/CI') since it is established automatically.

If the parameters for user name ('/ID'), password ('/PW') and perhaps the session ID ('/SI') have also been entered into the command line, the respective values are used when the session is established.

/OF

Ending a session
If a session is active, this parameter (after processing any parameter files) terminates it.

/SI x

Session ID for session
x => Session ID ('A' – 'Z')

If a 3270 emulation is used for the connection to the mainframe, the session ID to use can be defined with this parameter. Otherwise, the ID specified in the respective tcACCESS dialog (see Connection configurations, page 190) is used.

/ID xyz

User name for session

xyz => User name

This parameter allows you to define a user name that is used when the session is started.

/PW xyz

Password for session

xyz => Password

The password entered after the "/PW" parameter is transmitted to the host when a session is started.

/PF xyz

Parameter files to be processed

xyz => complete path of the parameter file to be processed

This command uses a parameter file or a recovery file that is saved in a tcACCESS dialog for the transfer of data.

In case of a parameter file, all values not defined explicitly are taken from the standard file (tcadef.xxx), which can be created using the parameter menu item "Set as default" of the respective dialog. If it does not exist, the program initializes all undefined values. In case of a recovery file, the settings of the abnormally terminated data-transfer and the status information will be loaded.

The following types of parameter files do exist:

- PC => Host Transfer (extension .pph)
- Host => PC Transfer (extension .php)
- Host SQL Query (extension .psq)

The following types of recovery files exist:

- Host => PC Transfer **(Extension ".rhp")**
- Host SQL-Query **(Extension ".rsq")**

For each tcACCESS call, only one parameter file or recovery file can be defined. If commands have been defined for establishing ('/ON') or deactivating ('/OF') a session, they are processed before or after this parameter.

/RC xyz

Changing the data transfer return code file

xyz => complete path of the return code file to be used for error output

If the data transfer return codes should not be written into the file tcaret.txt of the WORK directory, using this parameter may provide another file name. This is useful for multiple parallel data transfers (e.g., by the tcACCESS Listener) if the return code files are to be processed afterwards.

/NA

No "Cancel" button for data transfer

If a data transfer is initiated by using the parameter '/PF', this parameter allows you to suppress the display of the **Cancel** button in the data transfer status window. If the transfer has been completed successfully, the button changes to a Close button and is available again.

/NK

No key confirmation after a data transfer

In combination with automated processing (e.g., batch files), the need for any user input is generally not applicable. Therefore, this parameter allows you to automatically close the data transfer status window at the end of a transfer.

/NS

No status window during a data transfer

In order to perform a data transfer without the display of a status window, the '/NS' parameter is provided.

/QU

Close program after processing

This parameter ends the tcACCESS program when all other command line parameters have been processed.

/LIC xyz

Apply license file

xyz => fully qualified path to the license file

If a new license sholuld be applied, this parameter can be used. The correct license will be identified in the license file that corresponds to the license type (Administrator, Server, Custom) and will be applied. If a special license should be applied, the license type can be specified in addition to the fully qualified path. Example:

tcaccess /LIC C:\TEMP\SETUP.INI Server

or

tcaccess /LIC C:\TEMP\SETUP.INI License01

In addition, the following parameters are supported. They allow you to overwrite the values of the used parameter files:

PARAMETERS	DESCRIPTION
/BT	see /BT - Binary Data Transfer (page *336*)
/CP	*see /CP - VM/CP command to be processed* (page *338*)
/EK	see /EK - End Record or Number (page *340)*
/HF	see /HF - Host File name (page *342*)
/HP	see /HP - Host Server Module Parameter (page *346*)

PARAMETERS	DESCRIPTION
/LF	see /LF - PC File name (page *349*)
/MR	see /MR - Maximum Number of Records (page *349)*
/RS	see /RS - Record Selection (page *353*)
/RY	Specify the name and location of the recovery that should be created. If this parameter is specified, the option "Enable Recovery" is automaticly activated. **Example:** /RY C:\DATA\MyDownload.rhp
/SK	see /SK - Start Record or Number (page *355*)
/SQ	see /SQ - SQL Command (page *356*)
/SV	see /SV - Host Server Module (page *360*)
/XS	Defines the name and location of the XML schema file that should be created. If this parameter is specified, the option "Generate schema" is automaticly activated. **Example:** /XS C:\DATA\MySchema.xsd

Examples:

tcACCESS /SL 0 /ON /ID TESTUSER /PW TESTPASS

Establishing a host session using slot number 0 with a logon of user TESTUSER and his password TESTPASS.

TcACCESS /FSS /PF c:\temp\data.pph

Starting a data transfer from a PC to the host looking for an allocated but currently not used slot and using the parameters stored under data.pph.

tcACCESS /SL 5 /ON /ID TESTUSER /PW TESTPASS /PF c:\temp\data.pph /OF /NS /QU

- Establishing a host session using slot number 5.
- Processing the data transfer defined in the parameter file without displaying a status window.
- Deactivating the session.
- Terminating the tcACCESS program.

3. tcACCESS Request Monitor

3.1 General

The tcACCESS Request Monitor is a Snap-In program for the Microsoft Management Console (MMC). The tcACCESS Request Monitor can be used to monitor the overall tcACCESS communications and all of the ongoing data transfers.

3.2 Registration

Before a Snap-In can be used it must be registered. The registration of the tcACCESS Request Monitor is part of the tcACCESS Client setup and is being performed automatically. The following chapter explains how the registration and unregistration can be performed manually.

Enter the following commands into the Windows command box (Start/Execute):

To register:

```
regsvr32 /c C:\Program Files\tcACCESS\tcAMon.dll
```

To unregister:

```
regsvr32 /u /c C:\Program Files\tcACCESS\tcAMon.dll
```

The same directory path used for the installation of the tcACCESS client component should be specified.

3.3 Insert the tcACCESS Request Monitor Snap-In into the management console

After the start of the management console (mmc.exe) the following screen will be displayed:

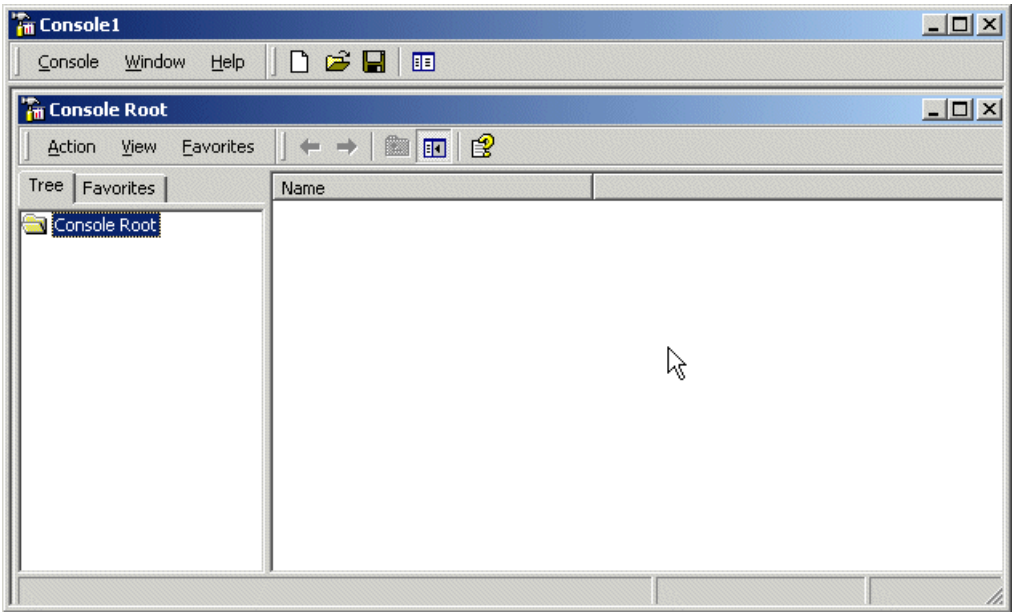


Fig. 190: Microsoft Management Console

Next, the tcACCESS Request Monitor Snap-In can be inserted. Select the menu item "Console/Insert-Remove Snap-In". The following dialog will be displayed:

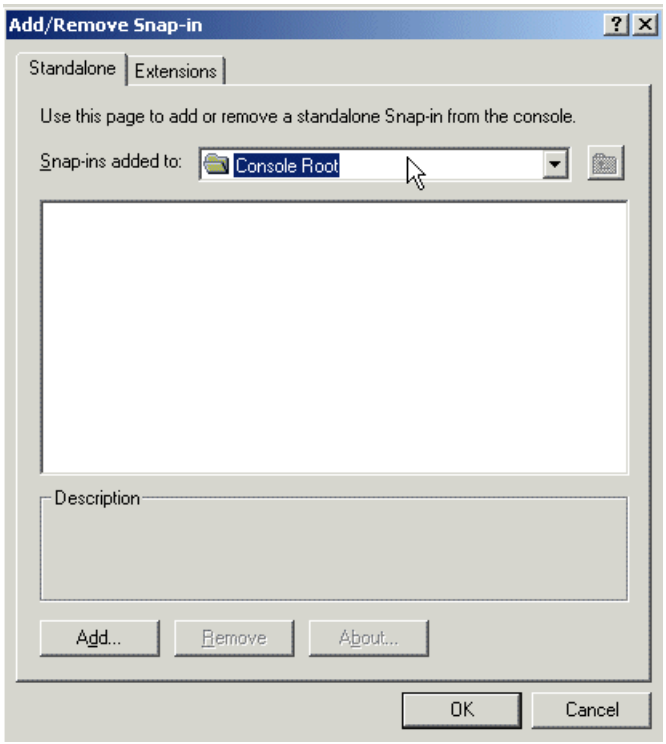


Fig. 191: Add/Remove Snap-In

Press the "ADD"-button and all available Snap-Ins will be displayed:

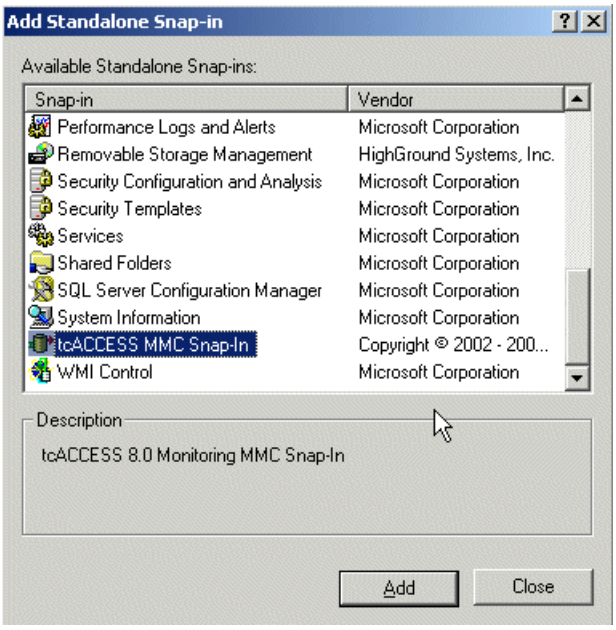


Fig. 192: Insert a Snap-In

Select the tcACCESS MMC Snap-In from the list and click the "Add"-button.The following dialog will be displayed:

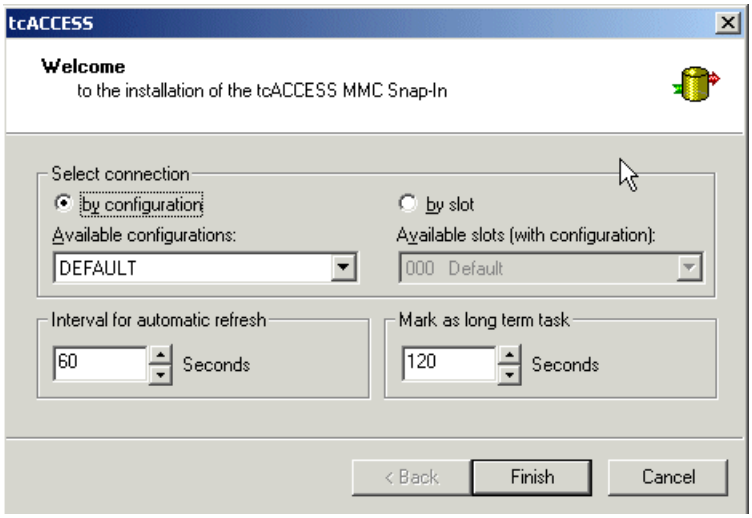


Fig. 193: Installation of the tcACCESS MMC Extension

The dialog options have the following meaning:

Select connection – by configuration

The configuration displayed will be used to establish a mainframe connection. Any slot assigned to this configuration will be used.

Select connection – by slot

A slot should be used to establish a connection with the mainframe. Select a slot from the list box (Available slots).

Interval for automatic refresh

The display of communications and data transfers are always snap-shots of the "at this second" situation. This entry specifies the interval in seconds when the console-display should be refreshed.

Mark as long term task

Communications and data transfers which have not performed a host access for a certain period of time will be indicated in the display (a red exclamation mark will be used). Specify the number of seconds that should expire until a task is being considered a "long term" task.

Finish

The dialog terminates and the snap-in will be added.

Cancel

Cancel the installation of the Snap-In.

After you have completed the installation of the Snap-In you can insert another Snap-In. We do not want to do this and click on the "Close"-button.

The "Add/Remove Snap-In"-dialog now displays the tcACCESS Snap-In.

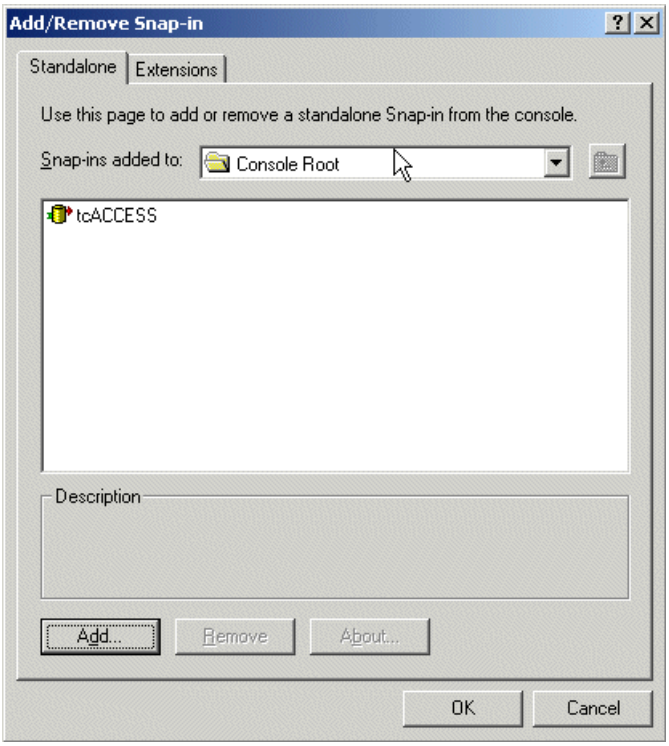


Fig. 194:Add/Remove Snap-In and the tcACCESS component

Click the "OK"-button and the tcACCESS Snap-In has been inserted in the console-base of the MMC.

3.4 Using the tcACCESS Request Monitor Snap-In in the MMC

The tcACCESS Snap-In has now been inserted into the MMC. This chapter will show you how to use the tcACCESS Request Monitor.

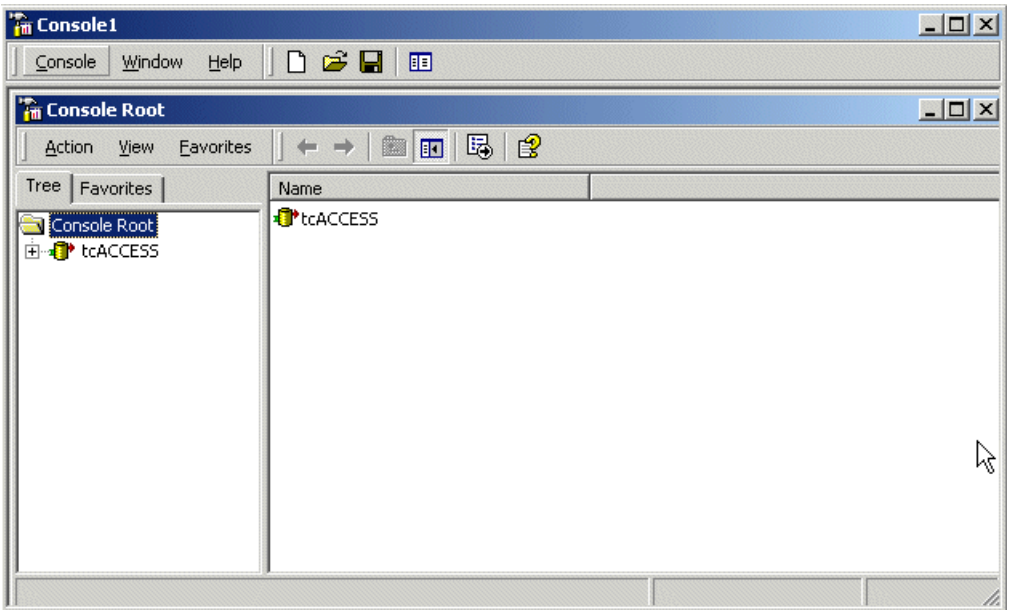


Fig. 195: MMC and the tcACCESS Snap-In

By simply clicking on the "Plus"-symbol in front of the tcACCESS entry, a connection to the mainframe will be established. For the connection either a configuration or an individual slot will be used. Remember? Thats what we specified when we installed the Snap-In. The tcACCESS Host Application Id will be displayed.

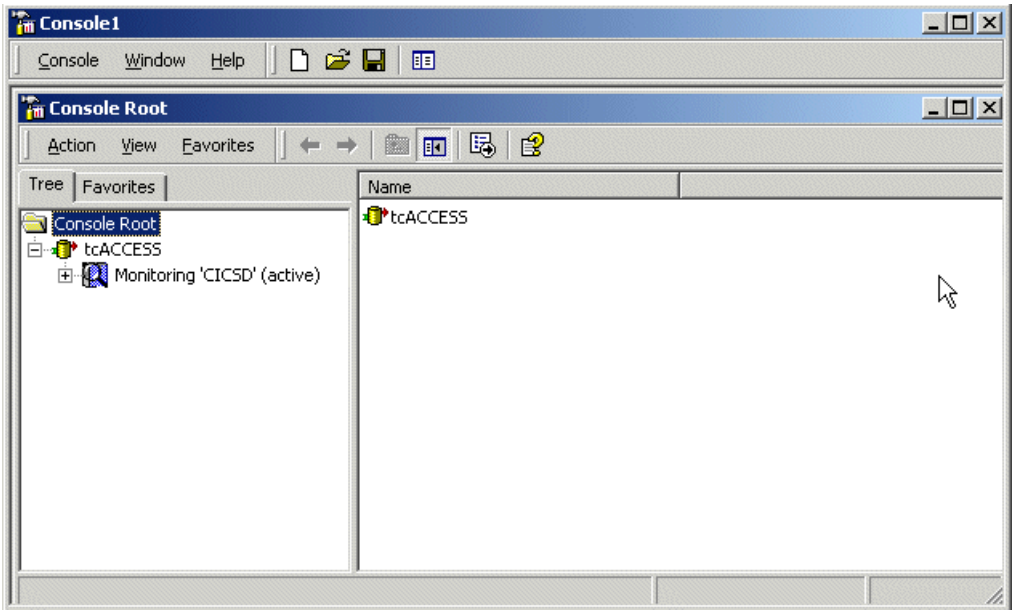


Fig. 196 MMC with active tcACCESS Request Monitoring

Press on the "Plus"-symbol in front of the monitor entry and all three levels of the monitoring will be displayed.

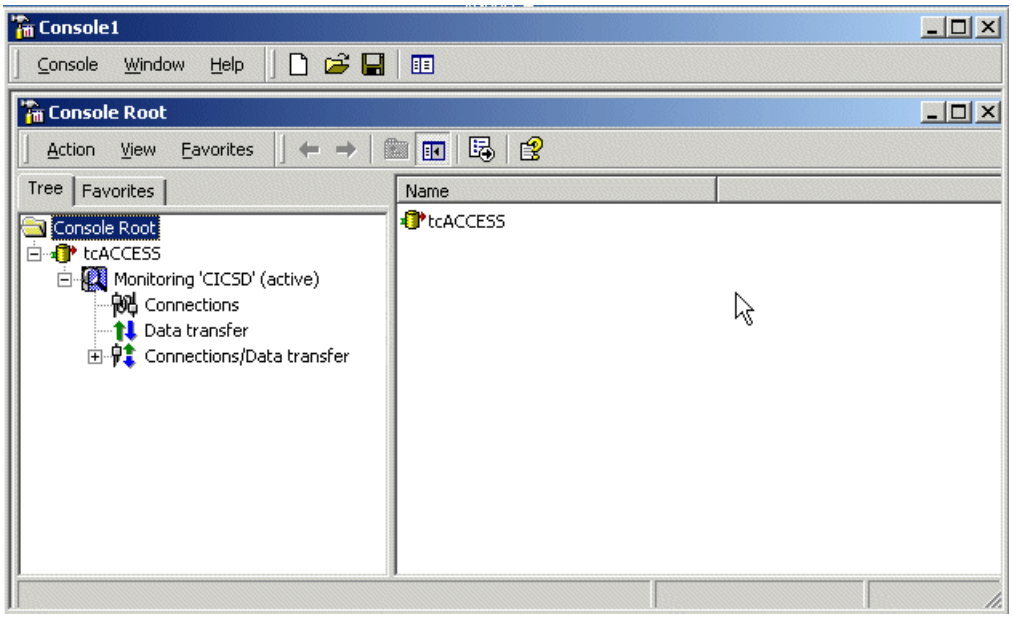


Fig. 197: MMC and the monitoring levels

By selecting the entry "Connections" all currently active connections to that host will be displayed. "Data transfers" Displays all existing data transfers between workstations and the the host. "Connections/Data transfers" display all the data transfers which are related to a connection.

3.4.1 Connection display

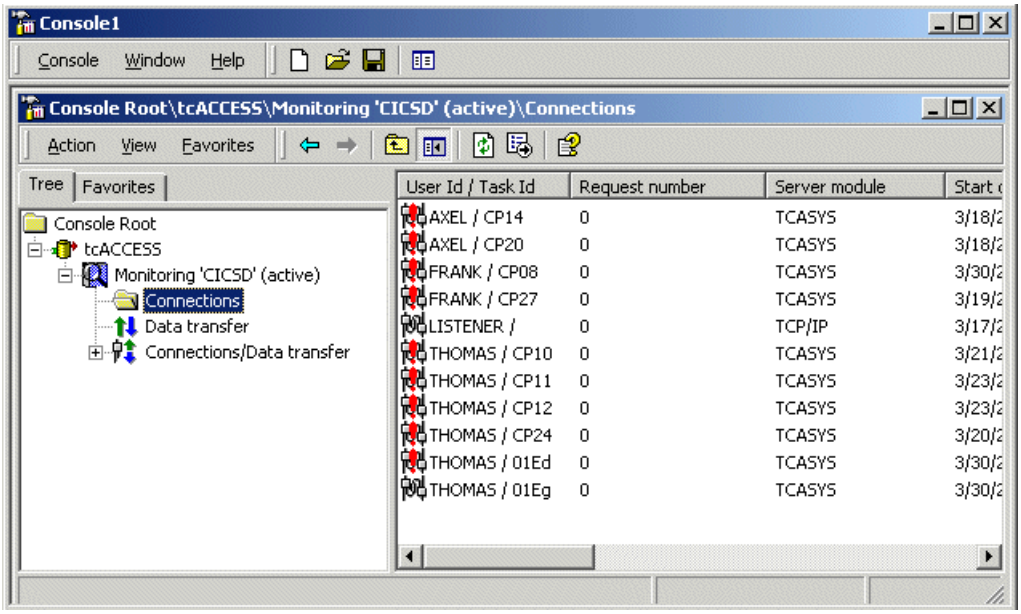


Fig. 198 MMC (Connection display)

On the right hand side all connections will be displayed which were active at the time of the last refresh. The following information will be displayed:

User Id / Task Id

The signed on user and the assigned task ID.

Request number

A communication can process multiple requests in parallel. To logically separate these requests an individual request number will be assigned.

Server module

The host server module that has been executed by the request.

Start of communication

Time when the connection has been established between the client and the host.

Last update of communication

Time of the last communication between client and host.

Actual function

The current function being performed by the host on behalf of the client.

Server call count

Running number of all calls to the host server module.

CICS task number

The task number of the CCS-system or the VTAM Monitor-system.

PC name

Name of the PC which hosts the connected client

The columns that are displayed and their sequence can be controlled by you. We'll cover this in section How to modify the console-display on page 288.

NOTE: A red exclamation mark in the connection symbol indicates, that the last host access for this communication has expired the number of seconds defined in the configuration for the Snap-In and that the tcACCESS Request Monitor treats this connection as being a "Long term"task. (refer to Insert the tcACCESS Request Monitor Snap-In into on page 278).

3.4.2 Display of data transfers

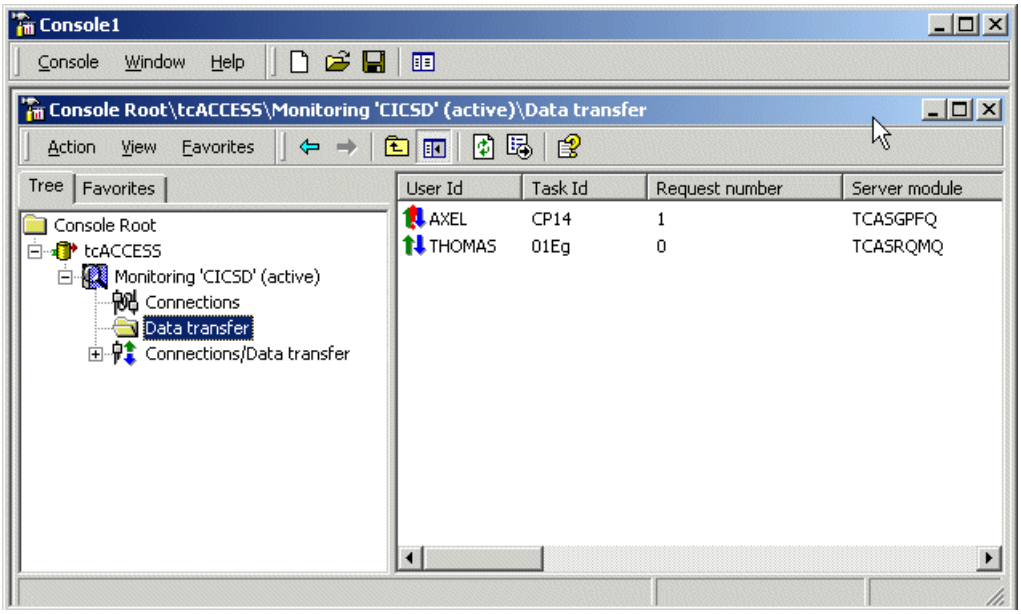


Fig. 199 MMC Display of data transfers)

On the right hand side all data transfers will be displayed which were active at the time of the last refresh. The following information will be displayed:

User Id / Task Id

The signed on user and the assigned task ID.

Request number

A communication can process multiple requests in parallel. To logically separate these requests an individual request number will be assigned.

Server module

The host server module that has been executed by the request.

Start of communication

Time when the connection has been established between the client and the host.

Last update of communication

Time of the last communication between client and host.

Actual function

The current function being performed by the host on behalf of the client.

Total IO count

The total number of IO operations for this data transfer.

Dead IO count

The number of IO which did not result in a record that was selected (i.e. selective data transfer, not selected records by the SQL-Engine).

Max. IO count

The maximum number of IO-operations that can be performed by this data transfer. If this threshold is reached., the transfer will be cancelled. The threshold is specified by either the –MAXIO statement or by the MAXIO parameter in the tcACCESS system generation.

Actual returned records

The number of records returned for this data transfer.

Max. records to return

The maximum number of records that can be returned by this data transfer. If this threshold is reached., the transfer will be cancelled. The threshold is specified by the "Max. records"-parameter for the SQL-Engine or the "Max. records"-parameter in the download definition.

Returned records for recovery

The number of the data-record that has been used for a restart..

Host file name

Name of the host resource that is processed (i.e. file name for an upload or download or a partial display of the SQL-statement that is currently being executed).

The columns that are displayed and their sequence can be controlled by you. We'll cover this in section How to modify the console-display on page 288.

NOTE: A red exclamation mark in the connection symbol indicates, that the last host access for this communication has expired the number of seconds defined in the configuration for the Snap-In and that the tcACCESS Request Monitor treats this connection as being a "Long term"task. (refer to Insert the tcACCESS Request Monitor Snap-In into on page 278).

3.4.3 Display of connections/Data transfers

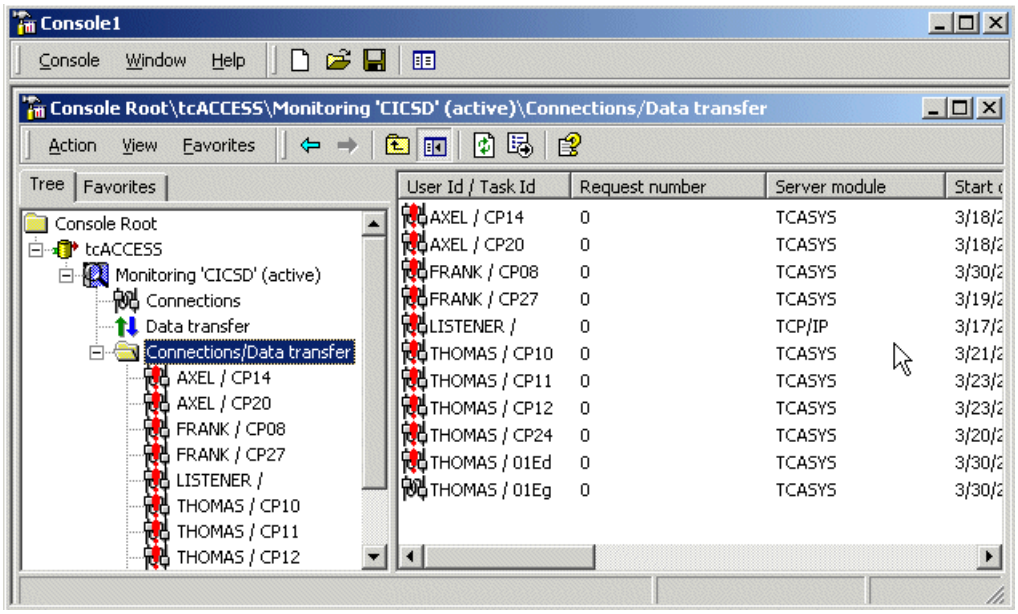


Fig. 200: MMC (Display of connections/data transfers)

The display on the right hand side is identical to the display of the connections.

By clicking on the connection entry in the tree view or by double clicking on a connection on the right hand side, the data transfers related to that connection will be displayed.

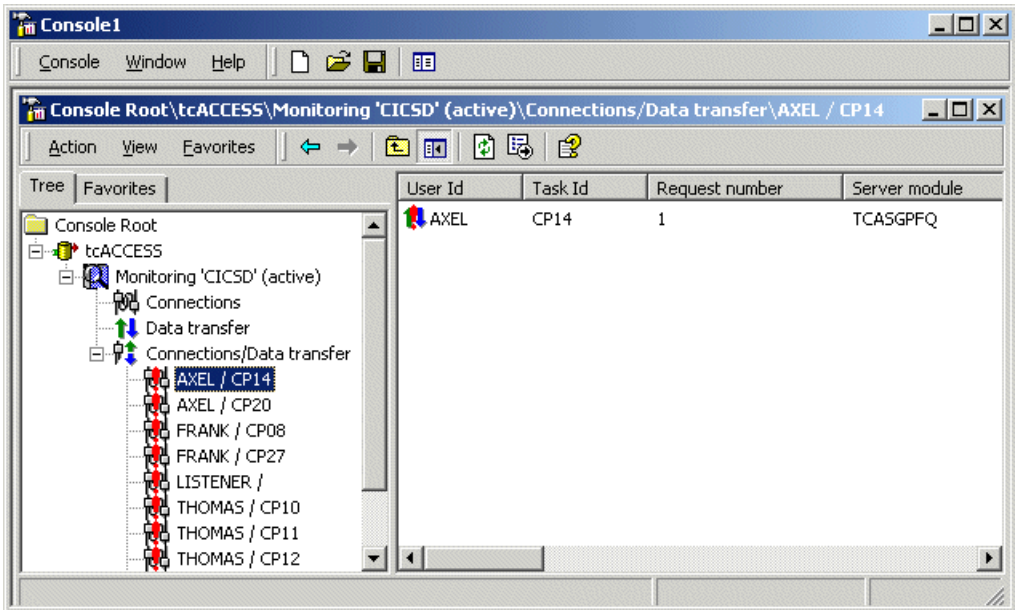


Fig. 201: MMC (Display of data-transers for a connection)

The display on the right hand side is identical to the display of the data transfers.

3.4.4 How to modify the console-display

If you want to adapt the information displayed to meet your personal requirements, select the menu item "Mofify columns" from the "View" menu. The following dialog will be displayed:

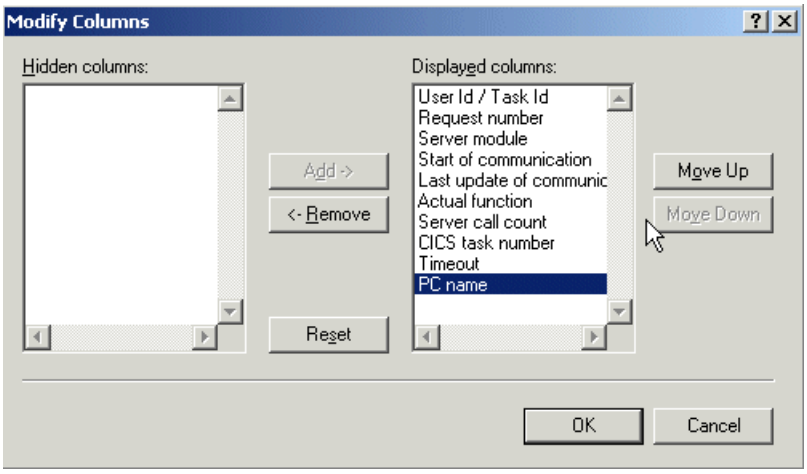


Fig. 202: MMC (Modify columns)

The dialog options have the following meaning:

Displayed columns:

The list shows all columns that can be displayed in the consol. The list can be rearranged, the sequence of the columns can be changed or columns can be deleted from the display (from top to bottom is identical to from left to right).

Columns User ID and Task ID cannot be rearranged and also cannot be excluded from the connection display. Column User ID cannot be rearranged and also cannot be excluded from the data transfer display.

Hidden columns:

The list shows all columns that are not displayed in the consol.

Add ->

Moves a selected column into the list of displayed columns.

<< Remove

Moves a selected column into the list of hidden columns.

Reset

Resets the display state.

Move Up

Moves a selected column up by one position in the list of displayed columns.

Move down

Moves a selected column down by one position in the list of displayed columns.

OK

Applies the changes and closes the dialog.

Cancel

Closes the dialog and no changes will be saved.

3.4.5 How to interact with the console

Context menus will be displayed if you right mouse click on the entries in the tree-view Some of these menus contain items that executed special functions of the tcACCESS Request Monitor.

3.4.5.1 Context menu for "Monitoring ..."

The context menu for the tree-view entry "Monitoring ..." is:

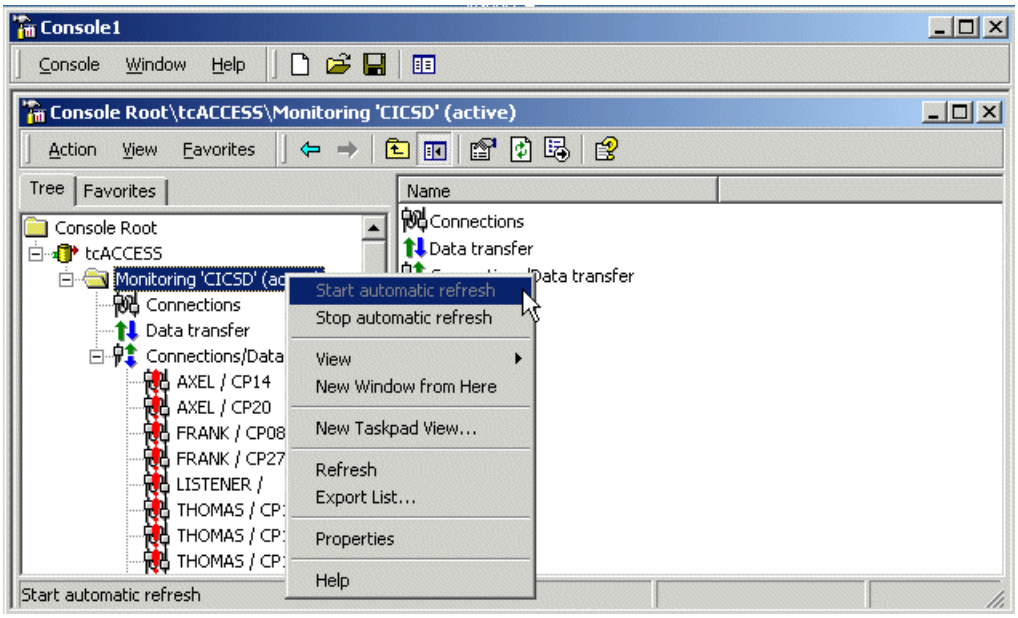


Fig. 203 MMC (Context menu for Monitoring)

Stop automatic refresh

This item can be used to stop the automatic refresh of the display. This item is only available if the automatic refresh is active.

Start automatic refresh

This item can be used to start the automatic refresh of the display. This item is only available if the automatic refresh is inactive.

Refresh

The display of the tcACCESS Request Monitor should be refreshed immediately.

Properties

The selection of this item will display the following dialog:

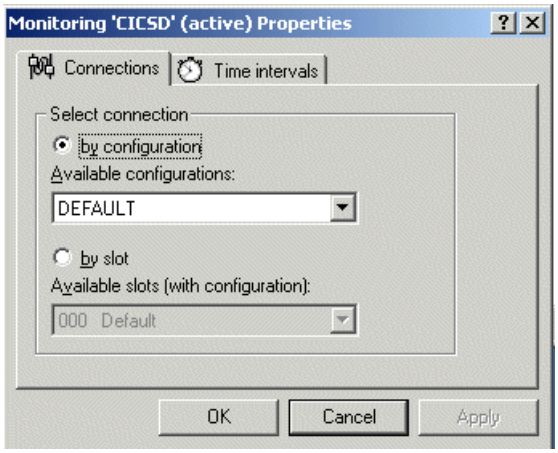


Fig. 204: Properties of Monitoring (Connections)

The dialog options have the following meaning:

Select connection – by configuration

The configuration displayed will be used to establish a mainframe connection. Any slot assigned to this configuration will be used.

Select connection – by slot

The specified slot should be used to establish a connection to the mainframe.

The second dialog looks like the following:

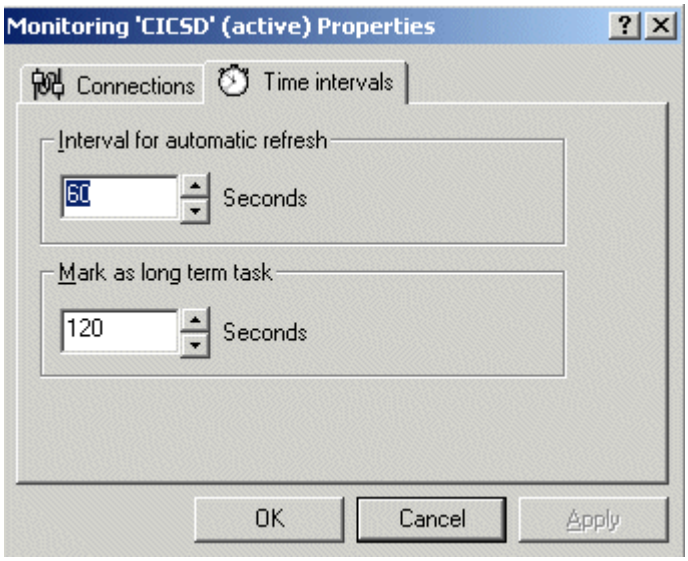


Fig. 205: Properties of connection (Time-intervals)

The options of the dialog have the following meaning:

Interval for automatic refresh

The display of communications and data transfers are always snap-shots of the "at this second" situation. This entry specifies the interval in seconds when the console-display should be refreshed.

Mark as long term task

Communications and data transfers which have not performed a host access for a certain period of time will be indicated in the display (a red exclamation mark will be used). Specify the number of seconds that should expire until a task is being considered a "long term" task.

OK

Applies the changes and closes the dialog. If a configuration or a slot have been changed the the old connection will be deallocated and a new connection will be established.

Cancel

Closes the dialog and no changes will be saved.

3.4.5.2 Context menu for "Data transfers"

The context menu for the entry "Data transfer" on the right hand side looks like the following:

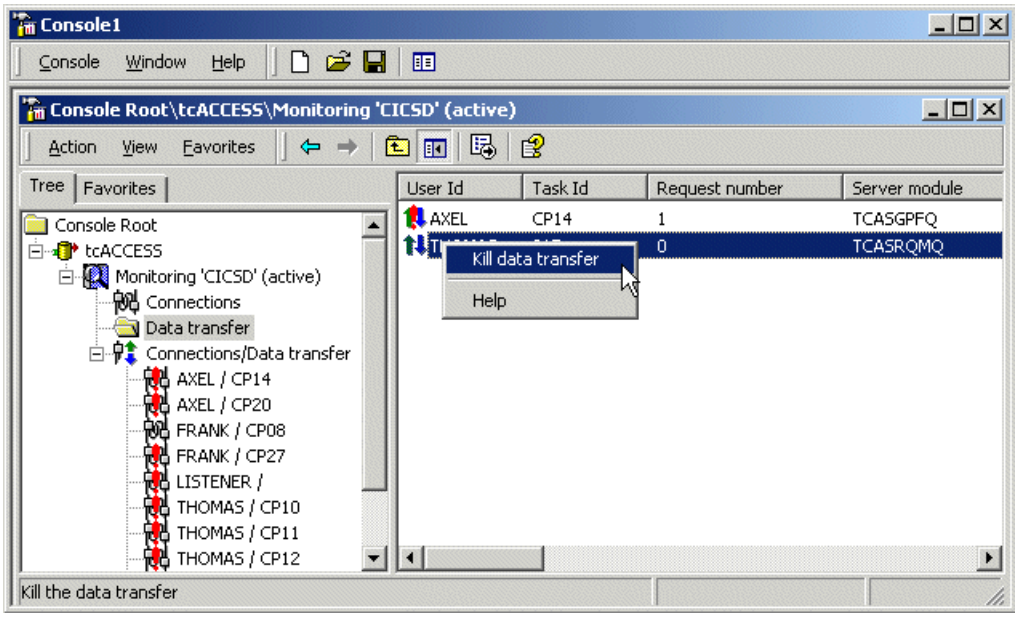


Fig. 206: MMC (Context menu for data transfers)

Kill data transfer

Using this menu item will cancel the data transfer. This may take some time. A data transfer that is marketd for cancellation but is not yet cancelled are indicated by an asterisk (*) in the User ID.

3.4.5.3 Context menu for "Connections"

The context menu for the entry "Connections" on the right hand side looks like the following:

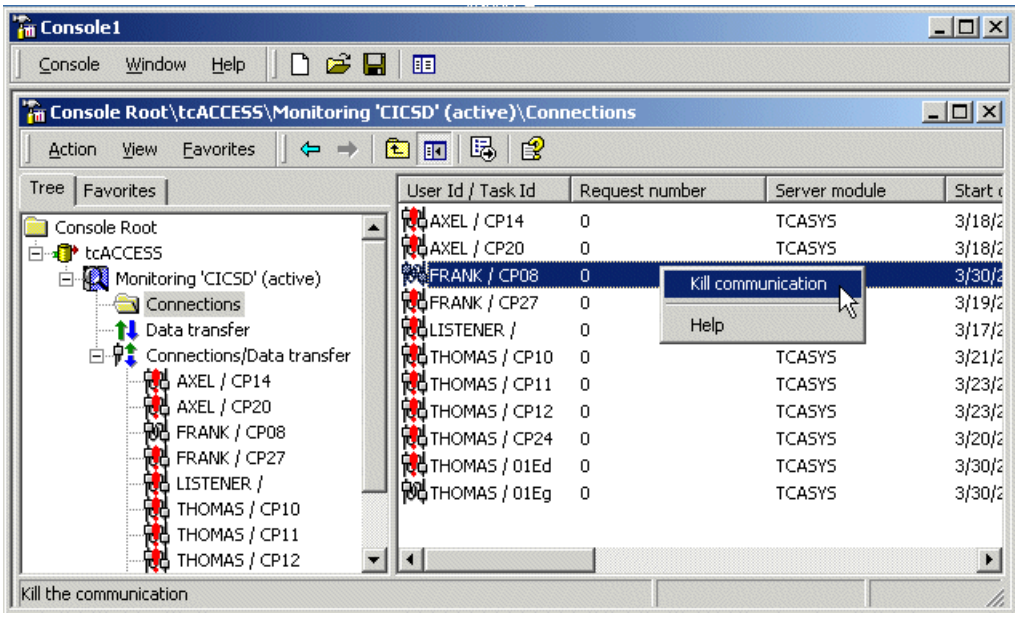


Fig. 207: MMC (context menu for connections)

Kill communication

Using this menu item will cancel the communication. This menu item will only be available if the parameter HNDABND has been set to "Y" in the tcACCESS system generation.

3.4.6 Save the console-settings

The console-settings should be saved so that the changes will be still active the next time you will use the console.

Select menu item "Console/Save as". Specify the file name in the dialog box.
During the next start of the console these settings can be loaded.

4. The tcACCESS Listener

The tcACCESS Listener allows you to perform host-initiated operations on a Windows PC. If you use TCP/IP, the system waits for messages from the host. If, on the other hand, you use the existing tcACCESS host connection, polling technology is used—the PC frequently sends requests to the host.

The tcACCESS Listener requires a 32-bit Windows (95/98, NT, 2000, XP, 2003) environment and is not executable under Windows 3.x.

After starting the TCALS.EXE program the main window of the application displays. It contains the menu to control the program and a list showing each processed operation and program messages.

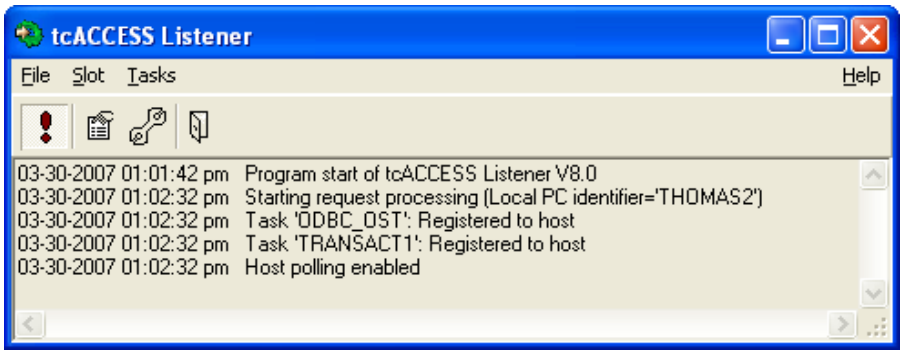


Fig. 208: tcACCESS Listener

Icon Bar

Using the icons in the icon bar frequently used functions can directly be started, without selecting the menu item. If the icon are grayed, they are not active (e.g. if no session with the host is established).

The icons are described below:

 .....Start task processing (refer to page 295)

 .....Stop task processing (refer to page 295)

 .....Process task list (refer to page 295)

 .....Process the setting (refer to page 299)

 .....Terminate the listener

4.1 The tcACCESS Listener Menu Structure

4.1.1 The File Menu

4.1.1.1 Change language

After selecting this menu item a list of the installed language modules displays.

Fig. 209: Change language dialog

The language currently used is marked when the dialog is called. After selecting another language and confirming with **OK,** the language for all tcACCESS displays changes.

4.1.1.2 Exit

This menu item ends the tcACCESS Listener. If processing is still in progress at this time, all active tasks are unregistered from the host before the program ends. If the TCP/IP Listener is started, it closes.

4.1.2 The Slot Menu

All slots assigned to a connection configuration are displayed here. A marked slot indicates the currently active slot. The active slot can only be changed if there are no active tasks.

The slots can be configured as described in chapter Connection configurations page 190.

4.1.3 The Tasks Menu

4.1.3.1 Start/stop request processing

The selection of this menu item starts or ends the processing of host requests. While it is running, the host registers all active tasks. When host-request processing ends, the registered tasks become unregistered.

Additionally, depending on the selected host connection for the task start (see Settings, page *299)* the system starts either a TCP/IP Listener or polling via the tcACCESS session.

After the request processing has been activated successfully, a '✓' icon displays in front of this menu item.

4.1.3.2 Task list

The window displayed after selecting this menu item contains all tasks defined by the user. The system saves each task in a separate parameter file within the TASKS sub-directory of the tcACCESS installation. You can also select this menu item when request processing is started. You can still use the menu items provided in the main window while this window is open.

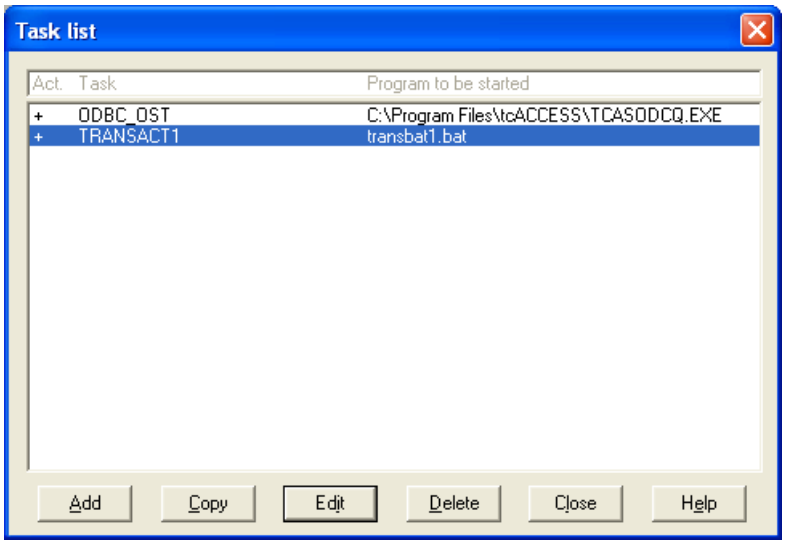


Fig. 210: tcACCESS Listener - Task list

The dialog components are described below:

List of tasks

The main part of the dialog contains the list of defined tasks. Each entry ready for host requests is marked with a '+' in front of it. You can activate or deactivate a task using the **Space** bar. If request processing is started at this time (see Start/stop request processing, page *295*), the related task immediately registers to or unregisters at the host. You can process a task using the **Enter** key (**Return**) and delete it with the **Del** key. You can add a new task using the **Ins** key (see Add/edit task on page *297*).

Add

Selecting this button opens an input window to describe a new task (see Add/edit task on page *297*). This has the same effect as pressing the **Ins** key within the task list.

Edit

Selecting this button opens a window with the current settings for the selected task (see Add/edit task on page *297*). The same effect results from pressing the **Enter** key within the task list.

Copy

Opens a window with a copy of the current settings for the selected task. The name of the copied task can be specified in the field "Task name".

Delete

If you want to remove a task permanently, press this button. Alternatively, you can use the **Del** key within the task list for this purpose. After confirmation, the system deletes the parameter file from the TASKS sub-directory and removes the task from the list.

Close

Closes the dialog window.

Help

Displays online help for this dialog.

4.1.3.2.1 Add/edit task

This dialog allows you to configure a task to be performed on the PC. The settings are saved in a parameter file within the TASKS directory.

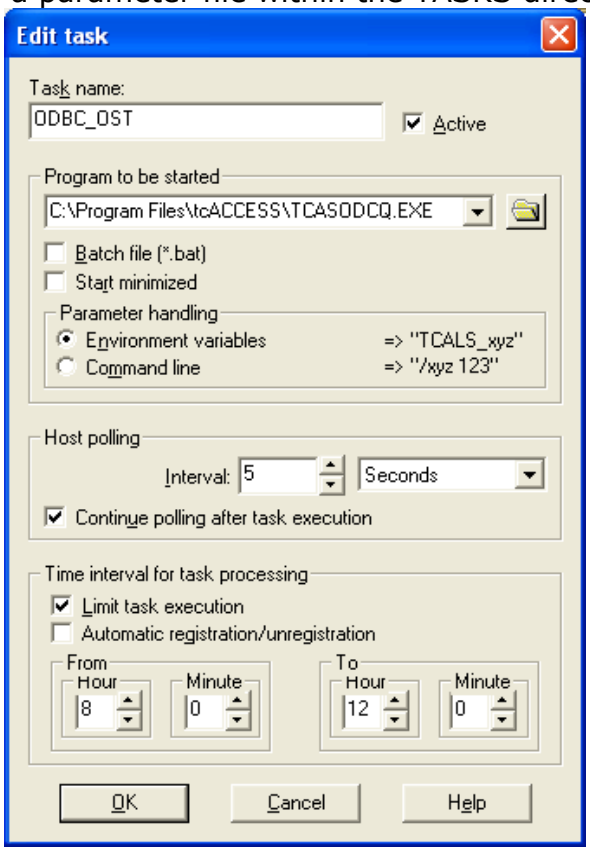


Fig. 211: tcACCESS Listener - Add/edit task

The dialog components are described below:

Task name

This field allows you to provide a name for the task, which must correspond with the request name passed from the host. This also becomes the name of the parameter file (e.g. C:\Program Files\tcACCESS\TASKS\my task.tsk).

Active

Click this checkbox if you wish to be able to start the task from the host.

Program to be started

The system executes this program or batch file as soon as the host starts the task. It passes several parameters (see Parameter handling). It may return an exit code which is then returned to the host as a result. It may also write a result text (up to 80 characters long) into the RESULTFILE file which is passed as a parameter and returned to the host.

Sample batch file:

```
echo Retreiving host data from TS queue %TCALS_HF%
%TCA%\tcACCESS /PF %TCAPARAM%\download.php /HF %TCALS_HF%
IF EXIST %TCAWORK%\tcaret.txt GOTO :ERROR
echo Download was successful!
EXIT 0
:ERROR
echo ERROR: Download failed!
copy %TCAWORK%\tcaret.txt %TCALS_RESULTFILE%
EXIT 1
```

Batch file (*.bat)

Select this button if the program to execute is a batch file. The batch file is started within a tcACCESS Command Shell. This method allows you to process commands sequentially and it supports the EXIT x batch command which returns an exit code 'x' (value between 0 and 255) to the tcACCESS Listener. Therefore, each exit code except zero represents an error which is then returned to the host with the hexadecimal offset 10B00000 (e.g. EXIT 5 results in the error code 10B00005 on the host).

Start minimized

Select this button if the program to start should not appear as a window but only within the task bar.

Parameter handling

There are two possibilities for passing parameters to the executing program:

- Environment variables - with TCALS_ prefix, e.g. TCALS_HF
- Command line - in the /parameter=value format, e.g. /HF=0002TCR0(TS)

If the program is a batch file, you should use environment variables set between percentage characters (e.g.: echo everything OK! > %TCALS_RESULTFILE%).

Interval for host polling

If a TCP/IP Listener does not handle the requests from the host, the PC must ask the host frequently if there is a task to be processed. The time

interval between two requests can be defined for each task individually by entering a value into this field. If you set the value to 0, the host is not polled.

Continue polling after task execution

This option decides whether to poll the host for a further request even if the task was already executed. If you do not use the option, the task is set to NOT ACTIVE after the first successful start.

Time interval for task processing

Limit task execution

Use this option to specify a time interval when the task can be executed. If an attempt is made to execute the task outside this interval, the attempt will be rejected and an error message is returned to the host.

Automatic registration/unregistration

If this option is active, taks will be automatically registered inside the specified time interval and will be automatically unregistered outside the time interval.
Note: No deregistration will be performed when the taks is still executing. In this case the unregistration will be performed after the task has ended.

OK

Pressing this button saves the settings in a parameter file and optionally registers or unregisters the changed task. The dialog then closes.

Cancel

Pressing this button closes the dialog without saving any modified task settings.

Help

Displays online help for the open dialog.

4.1.3.3 Settings

This dialog allows you to change general settings of the tcACCESS Listener.

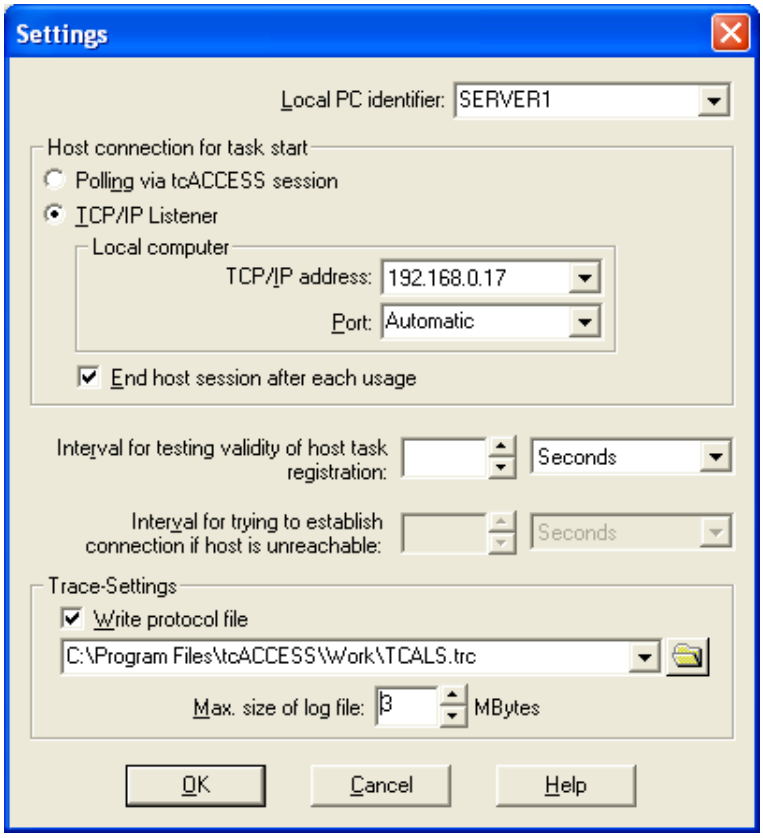


Fig. 212: tcACCESS Listener – Settings

The dialog components are described below:

Local PC identifier

For the registration of a task on the host there is always a logical name for the local PC needed beside the task name. If this input field is empty, the Windows PC name is used.

Host connection for task

The tcACCESS Listener provides two types of host connection.

If a TCP/IP host connection is used, it is helpful to select the **TCP/IP Listener** that waits on the PC until the host establishes a network connection and thus initiates the start of the task. For this connection type a free TCP/IP port must be defined on the local PC (default value: 2345). If the **End host session after each usage** button is active, the standard tcACCESS connection to the host is established for a minimum time only and immediately ended when it is no longer needed. If the connection has not been established by the tcACCESS Listener itself, this button is ignored and the connection remains.

The host connection of the type Polling via tcACCESS session causes more data traffic within a network but it has the advantage that this connection type operates with every tcACCESS Communication Handler (via 3270 emulation, APPC, MQSeries, ...). You can define the polling intervals individually for each task (see Add/edit task, page 297).

TCP/IP-address

Select the TCP/IP-address of the local PC-listener. This may be necessary when the PC has multiple network-cards and(or multiple TCP/IP-addresses. This is normally true for server-machine under Windows NT.

Port

By default the tcACCESS listener automatically determines a free port on the local PC. If a particular port should be used, it can be defined in this field. This may be helpful in a firewall situation, where every allowed TCP/IP port must be explicitly defined.

End host session afer each usage

With this option active the tcACCESS session will only be kept for the minimal needed time and then released. In case that the connection has not been established by the tcACCESS listener itself, this option will not be evaluated and the connection remains.

Interval for testing validity of host task registration

Using this field, it can be ensured that all registered tasks are still known to the host. A corresponding host request will be sent in the intervals defined and all tasks that are missing in the registration list will be registered

again. This may be necessary, when the CICS system on the mainframe is shut down on a daily base.

Interval for trying to establish connection if host is unreachable

If the test to check the task registration failed because the host was unreachable, the tcACCESS listener will attempt to reach the host in the defined intervals. This may be necessary when the CICS system is shut down for the night..

Trace settings

Write protocol file

If this option has been set, all activities of the listener will be logged to the specified file.

Max. size of log-file

This option limits the size of the log-file. The size definition is in megabytes (MB). If the log-file reaches the maximum size, the file will be renamed and "_BAK" will be appended to the filename and a new file will be created. If this new file hits the maximum size, the old backup file will be replaced by the current log-file and a new log-file will be created.
If the field is empty or a value of 0 has been defined the size of the file is not limited..

OK

Saves the changed settings and closes the dialog. If values have changed, you may be asked to restart the request processing after a confirmation.

Cancel

Ends the dialog without saving any changed settings.

Help

Displays online help for the open dialog.

4.1.3.4 Windows Service

The tcACCESS Listener can be configured as a Windows service (only Windows NT/2000/XP). This may be necessary for using the tcACCESS Listener on a server as a service doesn't need a user logon for execution.

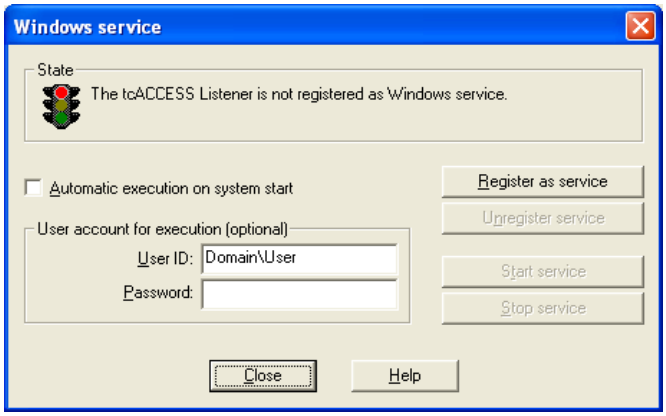


Fig. 213: tcACCESS Listener - Windows Service

The dialog components are described below:

State

This area shows the current state of the Windows service "tcACCESS Listener".

Automatic execution on system start

The "tcACCESS Listener" Windows service will be started on system boot automatically when using this option. Otherwise it has to be started manually. This can be done either by pressing the button "Start service" or by using the system dialog "Services" (Windows Control Panel).

Changing this option isn't possible if the service is already registered.

User account for execution (optional)

This input fields may be necessary if the execution of the "tcACCESS Listener" service needs access rights. A user account for running the service can be provided in this case. The domain name is also needed if the wanted user is authentificated by a Windows domain. It must be provided before the user name, separeted by a backslash ("DOMAIN\USER"). For example, the default access rights are insufficient when using a network drive (e.g. tcACCESS Client Installation).

Changing this option isn't possible if the service is already registered.

Register as service

This button evaluates the settings of "Automatic execution on system start" and "User account for execution" and then registers the tcACCESS Listener as a Windows service. Under Windows Vista the tcACCESS main program must be started as administrator to use this function.

Unregister service

The "tcACCESS Listener" Windows service will be removed from system by using this button. Under Windows Vista the tcACCESS main program must be started as administrator to use this function.

Start service

The "tcACCESS Listener" service can be started by using this button. A previous registration as a Windows service is mandatory for this action.

Stop service

A running "tcACCESS Listener" service may be stopped by using this button.

Close

Closes the dialog window.

Help

Displays online help for the open dialog.

4.1.3.5 Administration Server

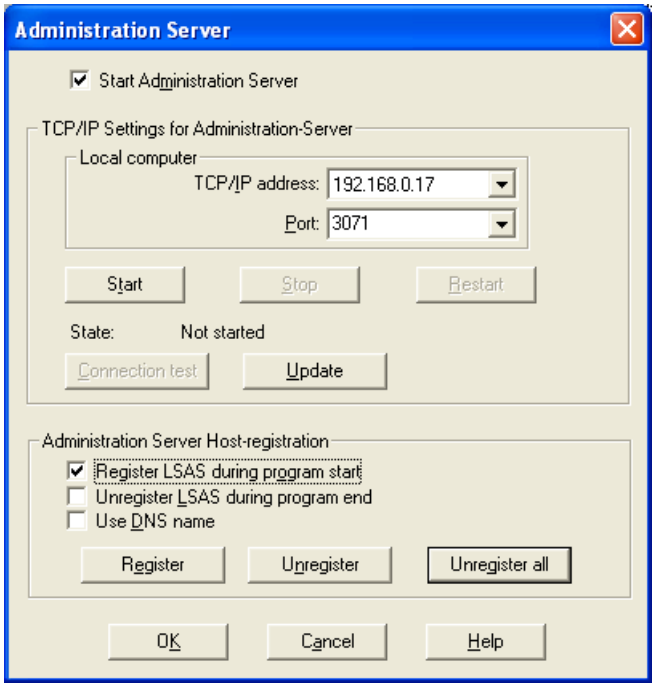
The administration server (LSAS) can be used to receive commands from the tcACCESS listener to control the task processing (i.e. start/stop taks processing, register/unregister tasks). The communication between an external program like the PC-batch utility BMLLS (refer to Command line program BMLLS of the tcACCESS-listener on page *310*) and the administration server is based upon TCP/IP.

The administration server can be contacted with the following commands:

- ⇨ **BMLLS** (refer to Command line program BMLLS of the tcACCESS-listener on page *310*)
  PC-command line program to control important tcACCESS listener functions (i.e. start/stop task processing, register/deregister individual tasks, check status).
- ⇨ **TCASLSTR** is a mainframe program whose intention is to inform the workstation listener when the host can be reached again (i.e. after a CICS restart). The task prcoessing of the listener will be started or restarted.

Note: It is only possible to start an administration server in one program instance of the tcACCESS listener. During initialization of the tcACCESS listener a check is performed whether the administration server has already been started by another instance of the tcACCESS listener (i.e.the tcACCESS listener as Windows-service has already been started with LSAS). In this case the administration server will not be started for the new instance of the tcACCESS listener.

Fig 214: tcACCESS Listener – Administration-Server



Start Administration Server

When this option is active the administration server will be automatically started during the program start of the tcACCESS listener. If this is not desirable, the administration server can also be started at a later time using the "START" button in the above dialog.

The automatic start of the administration server can be prevented (even with activated option "Start Administration server") by using the command line parameter **/NL** when starting the tcACCESS listener. (refer to *The tcACCESS Listener Command Line Parameter* on page *308*).

Note: In case the tcACCESS listener has been configured as Windows-service the option "Start Administration server" must be active, because a later start of the server for the Windows service is not possible..

TCP/IP settings for Administration-Server

Use this section to specify the address (TCP/IP address and port) where the adminisatration server can be reached. The administration server can also be manually started/stopped or the status of the administration server can be checked.

TCP/IP-address

Select the TCP/IP-address for the local PC-listener. This can be necessary when the PC has multiple network cards and/or multiple TCP/IP addresses. This is normally true for server machines under Windows NT.

Port

By default the administration server uses **Port 3071**. If another port should be used it can be specified here. If the input field is empty or 0 port 3071 will be used.

Start

Start the administration server.

Stop

Stop the administration server.

If the administration server has been started by another instance of a tcACCESS Listener (i.e. by a listener that runs as Windows-service) this "external" administration server will be stopped.

Note: If an external administration server is stopped that originally had been started by a listener service, this administration server can only be restarted by the service, when the service is stopped and restarted.

Restart

The administration server is restarted with the current settings for TCP/IP address and port.

If the administration server has been started in another instance of the tcACCESS listener (i.e. the listener Windows service) this "external" administration server gets restarted.

Status

Shows whether the administration service has been started or not. In addition the TCP/IP address and port used to communicate with the administration server is displayed. If the administration server has been started by another instance of the tcACCESS listener (i.e. the listener Windows service), it will be marked with "External".

Connection test

Use this button to thest a connection with an external administration server.

Update

Refresh the status display.

Administration server host-registration

The host registration informs the tcACCESS host component wich address to use to communicate with the administration server. The TCP/IP settings of the administration server (TCP/IP address and port) are saved to a file on the Virtual Disk. The name of the registration file is being built following the schema:

IP-Address_Port.TRIG or *DNS-Name_Port.TRIG* when option "Use DNS names" are active (**Examples**: 192.168.0.12_3399.TRIG or server1.comp1.com_3399.TRIG)

Usage: The host registration of the administration server enables the tcACCESS host component (program: TCALSTRG or TCVSTRG) to inform the tcACCESS listener that the host is "reachable" again, for example after a CICS-restart.

Note:

During the registration process the type of the host connection will be checked. If it is a TCP/IP connection, the registration file has a file suffix of ".TRIP", otherwise the suffix is ".TRIG". For registrations with suffix ".TRIP", a check will be performed during the start of the host components (programm TCALSTRG or TCVLSTRG) to determine whether the tcACCESS TCP/IP Listener is active on the mainframe. If the tcACCESS TCP/IP Listener is not active the process will be terminated. That means that the corresponding Administration Server will not be informed that the mainframe can be reached again.

Register LSAS during program start

When this option is active, the administration server will be automatically registered at the host during the program start of the tcACCESS listener.

Unregister LSAS during program end

When this option is active, the administration server will be automatically unregistered at the host during the program end of the tcACCESS listener.

Note: The LSAS will not be unregistered if the administration server is active in another instance of the tcACCESS listener (e.g. tcACCESS listener service) at program end.

Use DNS-names

When this option is active, the DNS name instead of the IP-address of the local computer will be used to construct the name of the registration (i.e. server1.comp1.com_3399.TRIG).

Register

Use this button to manually register the administration server at the host. The current settings of "TCP/IP address" and "Port" will be used.

Unregister

Use this button to unregister the administration server at the host. A registration will be removed from the host that relates to the current settings of the input fields "TCP/IP Address" and "Port" . If option "Use DNS-names" has been activated the registration will be removed that has been stored using the DNS-name and the specified port.

Unregister all

Use this button to unregister teh local administration server on the host. All registrations will be removed from the host that relate to the current settings of the input fields "TCP/IP Address" and "Port" . If option "Use DNS-names" has been activated the registrations will be removed that has been stored using the DNS-name and the specified port.

4.1.4 The Help Menu

4.1.4.1 tcACCESS Listener

Opens the online help for the tcACCESS Listener.

4.1.4.2 Content

Opens the online help function with a global overview display.

4.1.4.3 Index

Shows an index with the online help topics available.

4.1.4.4 About tcACCESS Listener

Contains information about the tcACCESS Listener.

4.2 The tcACCESS Listener Command Line Parameter

The command line parameters provided here allow you to set up automated processing of the the tcACCESS Listener.

The following parameters are supported:

/SL 123

Specify connection slot
123 => Slot number (between 0 and 254)
This parameter specifies the number of the connection slot. If no parameter specified to define a slot the tcACCESS Listener checks environment variable "DEFAULT_SLOT". This variable always exists, when a tcACCESS program has been started using the "tcACCESS Command Shell" (refer to The tcACCESS Command Shell BMLSHELL page *315)* and the variable contains a slot number. If this slot number does not exist, the slot number most recently used by the tcACCESS Listener for data communication between the PC and the mainframe will be used.

/MI

Start minimized
This parameter has the effect that the tcACCESS Listener appears in the task bar only but not as a window after program start.

/ID xyz

User name for session
xyz => User name
This parameter allows you to define a user name to use when starting the session.

/PW xyz

Password for session
xyz => Password
The password entered after the /PW parameter is transmitted to the host when a session starts.

/NK

No key confirmation necessary

If the system performs an automated start (e.g. during system start) user intervention should not be necessary. This parameter allows you to disable user intervention during automated starts.

/NL

Do not start administration server

Using this parameter no automatic start of the administation server during program start of the listener will take place.

/NR

Do not register administration server at host

Using this parameter no automatic registration of the administation server at the host during program start of the listener will take place.

/NU

Do not unregister administration server from host
Using this parameter no automatic unregistration of the administation server from the hostduring program start of the listener will take place.

/SP

Start request processing
During the program start of the tcACCESS Listener it is often necessary for the user to activate request processing from the host via the menu item provided. If you want this done automatically during program start, apply the /SP parameter.

/OF

Ending a session
If there is an active session, this parameter closes it when the tcACCESS Listener quits.

/TA 1.2.3.4

Specify local TCP/IP-address for Listener
1.2.3.4 => local TCP/IP-address

If thew local PC has multiple IP-addresses, using this parameter you can specify the IP-address to be used.

/TP 1234

TCP/IP-Port for local listener

1234 => TCP/IP-Port

The local IP port to be used, can be specified. This might be necessary, if only a certain port can be used (e.g. because of firewall requirements). If port 0 is specified, the Listener will automatically select a port number.

Example:

```
tcaLS /MI /SP /OF
```

The tcACCESS Listener starts with a minimized main window, host request processing activated and the existing host session ended at the end of the program.

4.3 Command line program BMLLS of the tcACCESS-listener

Using the command line program BMLLS maintenance calls of the local tcACCESS listener can be performed such as start/stop the task processing, registration/unregistration of tasks. BMLLS can also be used to obtain status information from the local tcACCESS-listener.

The returncode of BMLLS provide information about the execution results:
Returncode 0: successful program execution.
Returncode 1: Error during program execution. In case of an error a file with a name of "bmllsret.txt" will be created in the "WORK" directory of the tcACCESS installation.

This error file ist not created for returncodes that are a result of a status check because the returncodes are self-explanatory. Refer to the table of returncodes later on in this chapter.

The parameter to control the execution of BMLLScan be combined in any order. However, only ONE parameter to control the functions of an administration server and the corresponding sub-parameter can be passed. With other words only one function of the administration server can be called during a program execution of BMLLS.

Parameter to control the execution of BMLLS:

/CF xyz

File that contains an executable command-line

xyz => fully qualified path to the command-line file

The required command-lines can be saved to an ASCII file and the file name can be passed to BMLLS using the /CF parameter. The maximum size of the command line must not exceed 4000 bytes.

If the same parmater has been defined in the command-line file AND is also part of the command-line itself, the command-line parameter will be used.

Example: BMLLS /SC /NK /CF C:\temp\register.txt

Content of command-line file register.txt: /RT /TN TASK1,TASK2,TASK3

/NK

No key confirmation necessary

Using this parameter it is not necessary to manually close message windows that are displayed to document error conditions (refer to parameter /SC). The message windows automatically closes after a few seconds.

/RC filename

alternate filename for the error file

xyz => fully qualified path to the error file

In case of an error BMLLS creates a file that contains an error text. By default the filename for this file is "bmllsret.txt". The file will be created in the "WORK"-directory of the local tcACCESS installation.

If error messages should not be written into file "bmllsret.txt" in the "WORK"-directory another file can be spcified using this parameter.

Example: BMLLS /SP /RC C:\temp\bmllserror.txt

/SC

Show message window

By defaulot BMLLS does not display a message window in case of an error. Using parameter /SC an error message-window will be displayed.

/TO xyz

Maximum time to wait for a response of the administration server

xyz => maximum wait time in seconds

After a command has been sent to the administration server BMLLS will isse a default wait of 30 seconds to receive the reply. If no answer has been received within this timeperiod BMLLS will terminate with an error message. Parameter /TO can be used to define an alternate wait time for a response from the administration server.

/WT

Write logfile

This parameter creates a log file with a name of "BMMLS.TRC" in the "WORK"-dreictory of the tcACCESS installation. This option should be used if a need arises to provide additional information to the tcACCESS support team for problem resolution.

Parameter to call functions of the administration server:

/DT

De(Un)register task(s) from the host

Required parameter:

/TN xyz

Parameter /TN provides a list of tasks that should be unregistered from the host. The tasknames must be separated by commas.

Optional parameter:

/ND

By default tasks that are unregistered by BMLLS will also become inactive. An inactive taks will not be registered on the host during the start of the task processing. With parameter /ND the tasks will be unregistered from the host however their status remains "active". Please note, that tasks with a status of "active" will be registered again when option " Testing validity of task registrations on host" has been enabled.

Example: BMLLS /DT /TN TASK1,TASK2

This example unregisters tasks TASK1 and TASK2 and the tasks become inactive.

/GS

Check status

Per call of BMLLS with parameter /GS one of the following subparameter can be used for the status check.

Required parameter

/PS Status of taks processing

This parameter determines the current status of the task processing of the local tcACCESS listener. The status is represented by one of the following returncodes.

Returncode	**Status**
1	Error during program execution
2	Unknown status
3	Task processing is not active
4	Task processing has been stopped. Host currently not reachable.
5	Task processing is active

/OT xyz Status of a defined task

xyz => Taskname

Parameter /OT checks the status of the named task. The status is indicated in the returncode:.

Returncode	Status
1	Error during program execution
2	Unknown status
3	Task does not exist
4	Task is not active
5	Task is active
6	Task is registered but execution is currently not permitted.
7	Task is registered.
8	Task is currently executing.

Example: BMLLS /GS /OT TASK3

Determine the status of task TASK3 and return the status in the returncode.

/PP

Stop task processing

This parameter stops the task processing of the local listener. If tasks are still active the task processing will not be stopped and an error mesaage will be returned. This can be influenced by parameter /FS that forces to stop the task processing.

Optional parameter:

/FS

This parameter forces to stop the task processing, even if there are currently running tasks.

Example: BMLLS /PP /FS

/RT

Register task(s) on the host

Required parameter:

/TN xyz

/TN registers all defined tasks at the host. If multiple tasks are defined they must be separated by commas.

Example: BMLLS /RT /TN TASK1,TASK2

The example registers TASK1 and TASK2 at the host.

/SP

Start task processing

This parameter starts the task processing of the local listener..

/TL

Check availability of administration server

Use parameter /TL to check whether the administration server is running. If the administration server cannot be reached, BMLLS ends with an error message.

5. The tcACCESS Command Shell BMLSHELL

You can use the Command Shell to automate processes. It looks like an MS-DOS Command Shell and allows you to perform a number of similar operations, such as DIR, COPY, and TYPE commands. It also allows you to start programs and pass the output to a file, and to process batch commands. In addition, some Windows-specific commands are offered in order, for example, to close or hide a window.

5.1 Supported MS-DOS Commands

The listed commands are in general identical with the MS-DOS commands. They all support the /? parameter to display online help for the related command.

5.1.1 CALL

The CALL command allows you to start other batch files and waits for them ending before continuing execution (also allows you to provide parameters).

Example:

```
CALL batch2.bat param1 param2 param3
```

Note:

The CALL command must also be specified, if a batch file is to be started using the command line of the BMLSHELL.

5.1.2 CD

Change the current directory.

Example:

```
CD c:\temp
```

5.1.3 CLS

Clear the output window of a tcACCESS Command Shell.

5.1.4 COPY

Copy files and rename them as desired. The tcACCESS implementation is equivalent to the MS-DOS command of the same name, except that it does not support the /b parameter and the join of several files with the '+' operator.
In addition, the wildcards '*' and '?' are supported. You can copy files from or to virtual disks on the host (see Virtual Disks, Page 179).

Example:

```
COPY c:\*.txt s:
```

5.1.5 DEL, ERASE

Delete files. The specification may contain wildcards '*' and '?'.

Example:

```
DEL s:\text.doc
```

5.1.6 DIR

Display contents of directories. The /p parameter for page-separated output is also supported. If a UNC name is specified, all handles of the resource will be displayed.

Example:

```
DIR c:\temp
DIR \\dts
```

5.1.7 ECHO

Displays the character string following it on the screen. ECHO ON and ECHO OFF toggle the display of each batch command on the screen. If ECHO is OFF, the output of ID= and EC= are suppressed.
The tcACCESS Command Shell also supports the diversion of the ECHO output into a file using > or >>.

Example:

```
@ECHO off
ECHO line 1 > output.txt
ECHO line 2 >> output.txt
```

5.1.8 EXIT

Closes the tcACCESS Command Shell. If it is desired or necessary, a return code can be specified as parameter.

Example:

```
EXIT 4
```

5.1.9 FOR

Creates self-terminating program loops, and operates in the same manner as the MS-DOS command with this name.

Example:

```
FOR %%N in (*.DAT) DO CALL UPLOAD.BAT %%N
```

5.1.10 GOTO

Creates program branches to labels. Labels must start with a colon.

Example:

```
GOTO :label2
ECHO This line is not executed
```

```
:label2
ECHO this is executed
```

5.1.11 IF

Evaluates conditions and executes the related parameter when the condition is true. It operates like the MS-DOS command with the same name and supports the following types of conditions:

- Verify if a file exists (IF EXIST c:\autoexec.bat ...)
- Verify if ERRORLEVEL variable greater or equal value (IF ERRORLEVEL 4 ...)
- Comparison of expressions (IF %1 = = tcaccess ...)
- Verify if a directory exists (IF ISDIR C:\ABC …)
- Verify if a window exits (IF EXISTWINDOW 'Excel' ...)

The name to search for can also be a part of a window name.

Example:

```
CLS
IF %1= =excel GOTO :excel1
ECHO The first parameter must be excel
GOTO :end

:excel1
IF EXISTWINDOW 'Microsoft Excel' ECHO Excel is already
active!
IF NOT EXISTWINDOW 'Microsoft Excel' START
c:\excel\excel

:end
IF NOT EXIST c:\autoexec.bat ECHO By the way, you have
no AUTOEXEC.BAT!
```

5.1.12 MD, MKDIR

Create a sub-directory.

Example:

```
MKDIR c:\temp\data
```

5.1.13 PAUSE

Waits for a key-press before continuing execution of the batch file

5.1.14 RD, RMDIR

Remove an empty sub-directory.

Example:

```
RMDIR c:\temp\data
```

5.1.15 REM

Treats all following characters as non-executing comments. Several special commands of the tcACCESS Command Shell are excepted. They are described under REM $$ (page 324).

5.1.16 REN, RENAME

Change file and directory names.

Example:

```
RENAME c:\temp\test.dat *.bak
```

5.1.17 SET

Add an entry to a local environment or to change an existing entry. However, you cannot change any entries that were set before running Windows (e.g., the PATH variable).
The environment variable modified using the SET command no longer exists when the respective tcACCESS Command Shell is closed. Only the Shell itself and programs called from the Shell can use them.

Example:

```
SET progparm=c:\autoexec.bat
TYPE %progparm%
SET progparm=
```

5.1.18 START

Processes a program without waiting for its end. If an application is started without this command, the tcACCESS Command Shell (or the batch file processing) waits for the end of the called program in order to save the received return code in the ERRORLEVEL variable.

Example:

```
START c:\excel\excel
```

5.1.19 TYPE

Edit the content of text files within the tcACCESS Command Shell.

Example:

```
TYPE c:\autoexec.bat
```

5.2 Extended Commands

The following listing shows all extensions that are only valid for the processing within the BMLSHELL. They all support the /? parameter to display online help for the related command.

5.2.1 CLIP

Executes all lines of the current clipboard content as individual commands.

Example:

1. Start the Windows text editor (NOTEPAD.EXE)
2. Input the following lines:

```
c:
cd \temp
dir
```

3. Mark these lines and copy them into the clipboard (e.g. with **CTRL** + **C**)
4. Start the tcACCESS Command Shell BMLSHELL.EXE
5. Immediately after entering the CLIP command the command sequence in the clipboard area is processed in the tcACCESS Command Shell.

5.2.2 COMPUTE

Command "COMPUTE" represents an internal calculator with Infix-Notation.

The following operators are supported:
0x for sedecimal numbers, with leading for octale numbers, decimal numbers.

The following mathematical operations are supported:
brackets (), ~ (complement), * (multiply), / (divide), % (mod), + (plus), - (minus), L and R (shift left and right, & (bit AND), ^ (bit XOR), | (bit OR).

The calculator must be used with integer values. Simple operations can be performed during batch processing by using environment varaibles.

Example:

```
Example with different slot usage:
SET SLOTNR=0
… /SL %SLOTNR%
COMPUTE %SLOTNR% + 1
SET SLOTNR=%COMPUTE%
GOTO :BEGINNG
```

5.2.3 DIALOG_...

Allow simple user dialogs, which may contain select buttons, text input fields or the output of messages. Further, it is possible to use another button for **Cancel** in addition to the **OK** button.

The following environment variables control the execution of the DIALOG_xxx commands:

VARIABLE	DESCRIPTION
DLG_TITLE	Dialog title
DLG_OK	Text for **OK** button (Default: OK)
DLG_CANCEL	Text for **Cancel** button (Only created when this variable exists.)
DLG_CTRLy	Description for input field y y is a number between 1 and 4
DLG_TEXTy	Input and output text of the input field y (only for DIALOG_EDIT) y is a number between 1 and 4

If one of the four DLG_CTRLy variables does not exist, the respective input field is not available.
After the end of the respective dialog, the ERRORLEVEL variable is set as follows:

SETTING	DESCRIPTION
99	The dialog was terminated or closed.
4	While pressing OK the selection button 4 has been selected (only for DIALOG_RB).
3	While pressing OK the selection button 3 has been selected (only for DIALOG_RB).
2	While pressing OK the selection button 2 has been selected (only for DIALOG_RB).

1	While pressing OK the selection button 1 has been selected (only for DIALOG_RB).
0	The dialog was ended with OK.

5.2.3.1 DIALOG_EDIT

Opens a dialog with up to four input fields. Each of the DLG_CTRLy environment variables must contain descriptive text. If it is missing, the related input field is blocked.

The content of the input fields is provided in the DLG_TEXT1 to DLG_TEXT4 environment variables. If they are set before the dialog is opened, the input are preset with these values. An exception is the PASSWORD value, which may be used to create an input field for the password (no screen output).

Example:

```
...
SET DLG_TITLE=User logon
SET DLG_OK=OK
SET DLG_CANCEL=cancel
SET DLG_CTRL1=User name:
SET DLG_TEXT1=
SET DLG_CTRL2=Password:
SET DLG_TEXT2=PASSWORD

DIALOG_EDIT

IF ERRORLEVEL 99 GOTO :cancel
REM Info: DLG_TEXT1 contains the user name, DLG_TEXT2
the password
...
```

5.2.3.2 DIALOG_MSG

Creates a window for the display of a message. It must be specified in the DLG_CTRL1 environment variable before the call is made.

Example:

```
...
SET DLG_TITLE=Message
SET DLG_OK=OK
SET DLG_CANCEL=
SET DLG_CTRL1=This message is now displayed!

DIALOG_MSG

IF ERRORLEVEL 99 ...
...
```

5.2.3.3 DIALOG_RB

Creates a dialog with up to four selection buttons. The DLG_CTRL1 to DLG_CTRL4 environment variables must contain a description for each button. Otherwise the system does not create them.

Example:

```
...
SET DLG_TITLE=Please select
SET DLG_OK=OK
SET DLG_CANCEL=Cancel
SET DLG_CTRL1=Option 1
SET DLG_CTRL2=Option 2

DIALOG_RB

IF ERRORLEVEL 99 GOTO :Cancel
IF ERRORLEVEL 2 GOTO :opt2
IF ERRORLEVEL 1 GOTO :opt1
...
```

5.2.4 ENV_LEFT

This command sets a given environment variable to the specified length. The value of the length must be provided comma separated after the name of the environment variable.

Example:

```
SET MYVAR=EXAMPLE_TEXT
ENV_LEFT MYVAR,5          ⇒    %MYVAR%    =
EXAMP
```

5.2.5 ENV_LOWER

This command converts a given environment variable to lower case.

Example:

```
SET MYVAR=EXAMPLE_TEXT
ENV_LOWER MYVAR           ⇒    %MYVAR% = example_text
```

5.2.6 ENV_MID

This command cuts out a part of an environment variable. The name of the variable must be followed by the start position and the length of the chosen section. The three values have to be separated by commas.

Example:

```
SET MYVAR=EXAMPLE_TEXT
ENV_MID MYVAR,3,4         ⇒    %MYVAR% = AMPL
```

5.2.7 ENV_RIGHT

The command "ENV_RIGHT" sets an environment variable to a specified number of characters by removing characters from the beginning of the character string until reaching the desired length. The length has to be separated by a comma from the name of the environment variable.

Example:

```
SET MYVAR=EXAMPLE_TEXT
ENV_RIGHT MYVAR,3        ⇒    %MYVAR% = EXT
```

5.2.8 ENV_UPPER

This command converts a given environment variable to upper case.

Example:

```
SET MYVAR=Example_Text
ENV_UPPER MYVAR          ⇒    %MYVAR%          =
EXAMPLE_TEXT
```

5.2.9 EDIT

Starts the Windows text editor (NOTEPAD.EXE) with the defined parameters.

Example:

```
EDIT c:\autoexec.bat
```

5.2.10 GSET

Allows you to add variables to the local environment and to the global tcACCESS environment (saved in TCACCESS.INI). Thus, they are available again if a tcACCESS Command Shell is restarted.
Environment variables defined before running Windows must not be changed (e.g., PATH variable).

Example:

```
GSET TCAUSR=c:\temp
TYPE %TCAUSR%\data.txt
```

5.2.11 GSUBST

Connects a free drive letter with a virtual disk on a mainframe. The syntax is comparable with the MS-DOS command SUBST. The settings defined with the GSUBST command are permanently saved in the TCACCESS.INI file.

Examples:

```
GSUBST S: TCAVDI.SYSDSK
```

Connecting drive letter 'S' with the virtual disk TCAVDI.SYSDSK

```
GSUBST S: /d
```

Separating the connected drive letter 'S'

```
GSUBST
```
Display of currently connected drives

5.2.12 ISDIR

"ISDIR" checks whether the specified argument is a directory. This command can be used for iterative or recursive processing on a file or directory level.

Example:

```
IF NOT ISDIR DIR1 GOTO BEGINNG
```

5.2.13

5.2.14 LSET

Loads the currently defined global environment variables from the TCACCESS.INI. Thus, several active tcACCESS Command Shells can exchange data between each other as necessary.

5.2.15 LSUBST

Reestablishes connections between drive letters and virtual disks that were created by GSUBST calls. After starting the tcACCESS Command Shell LSUBST is executed automatically.

5.2.16 REM $$...

For those commands beginning with REM $$ requiring a window name as a parameter, it is not necessary to use the complete name of the target window. It is sufficient that the value matches with a part of it.

REM $$close
Closes the defined window.

Example:

```
REM $$close Microsoft Excel
```

REM $$delay
Delays the execution of the next command by 'n' milliseconds.
Example:

```
REM $$delay 4000
```

REM $$hide
Hides the window defined as parameter. If none is specified, the command hides the tcACCESS Command Shell itself.

Example:

```
REM $$hide Microsoft Excel
```

REM $$minimize
Minimizes a window. If no window name is defined as parameter, the command minimizes the tcACCESS Command Shell itself.

Example:

```
REM $$minimize Microsoft Excel
```

REM $$restore
Processes the Windows command 'Restore' for the defined window. If there is no window specified, the command restores the tcACCESS Command Shell itself.

Example:

```
REM $$restore
```

REM $$startminimized
Causes all Windows applications to appear in minimized form as a selection on the taskbar.

REM $$startnormal
Causes all Windows applications to display as windows in their standard size.

REM $$title
Transforms the name of the window for the tcACCESS Command Shell to the specified parameter value.

Example:

```
REM $$title Data transfer to host
```

REM $$unhide
Redisplays a hidden window. If you do not supply a window name parameter, the command redisplays the tcACCESS Command Shell.

Example:

```
REM $$unhide Microsoft Excel
```

5.2.17 SUBST

Connects a drive letter with a virtual disk within the current tcACCESS Command Shell (see GSUBST, page 323). Any global connections (created with GSUBST) have a lower priority and are temporarily overwritten with the SUBST specification. The connection between the drive letter and the virtual disk is only active during runtime and solely within the tcACCESS Command Shell.

5.2.18 WINDOWLIST

Displays a list of all active applications of the current Windows environment.

Example:

```
WINDOWLIST > wlist.dat
```

5.3 Automatic Insertion of Current Clipboard Content

As soon as any Windows application copies a text string containing $$tcaccess in the first line to the clipboard while the Command Shell is running, the system executes the rest of the clipboard contents as commands in the tcACCESS Command Shell.

Example:

1. Start the tcACCESS Command Shell BMLSHELL.EXE.
2. Start the NOTEPAD.EXE Windows editor and enter the following lines:

```
$$tcaccess
dir
```

3. Mark these lines and copy them into the clipboard (e.g. with **CTRL** + **C**).
4. Directly after copying to the clipboard the system executes the DIR and the contents of the current directory displays in the tcACCESS Command Shell.

5.4 Command Line Parameters

The tcACCESS Command Shell supports the following command line parameters:

/SL 123

Specify connection slot
123 => Slot number (between 0 and 254)
This parameter defines the number of the connection slot to be used. If this parameter has been omitted, the slot used to establish the connection for BMLSHELL when it was executed the last time will be used.

/HI

Hide tcACCESS Command Shell
Do not display as either a window or a taskbar icon.

/MI

Minimize program window
Display only as a taskbar icon.

/TI xyz

Set window title
xyz => window title
Customizes the title of the tcACCESS Command Shell window.

/C xyz

Execute command
xyz => command
Executes the supplied command after the program start. If you wish to use this parameter, it must always be last in the list of parameters.

/NC

No program close after command execution
Closes after executing the /C command by default. The parameter /NC can be used to prevent this behavior.

Example:

```
BMLSHELL /MI /C call test.bat
```

5.5 Predefined Environment Variables

The user can modify the contents of the predefined environment variables for the tcACCESS Command Shell with the SET and GSET commands.

The following environment variables are available:

VARIABLE	DESCRIPTION
TCA	Path to tcACCESS base directories
TCAUSR	Path to user-specific files (optional; after installation identical to the TCAWORK variable)
TCAPARAM	Path to the tcACCESS parameter files
TCAWORK	The tcACCESS work directory defined during installation
SYS_TIME	Current system time in the format hh.mm.ss
SYS_DATE	Current system date in the format yyyy-mm-dd.
DEFAULT_SLOT	Number of the currently used connection slot

6. The PC Communication Handler: "BMLCH"

6.1 The PC Communication Handler as executable module

Calling program "BMLCH.EXE" will start the PC Communication Handler. This component controls all data transmissions between the PC and the host using a connection slot. A windows is used to display certain information.

If the "TCP/IP TELNET 3270" Communication Handler is being used, you can open window "tcACCESS TELNET 3270", which is a full 3270 Telnet emulation. The emulation can be bused to establish a session with the mainframe.

Sub menu "Slot" displays all slots connected with a configuration. The connection status is displayed. It is possible to display a 3270 Telnet window and cancel the connection. tcACCESS tries to disconnect the session. If this cannot be accomplished within 4 seconds, the connection is aborted.

When started, the PC Communication Handler does not display an icon in the task bar but rather identifies as a small symbol at the right hand side. Left mouse click on that symbol displays a context menu. You can use this menu to disconnect certain slot connections.

If parameter /MI has been used when starting the communication handler, the communication handler will be started minimized, even if the diagnostic window has been activated.

6.2 The PC Communication Handler as Dynamic Link Library (dll)

Optionally, the tcACCESS Communication Handler can be loaded as a Dynamic Link Library (dll) by an application (example: tcACCESS Client, tcACCESS ODBC application). This option is highly recommended when tcACCESS (example: the tcACCESS ODBC driver) s executed as part of a server based application (i.e. WEB application, MSSQL-Server, etc.) or when tcACCESS is used by multiple users in parallel on a server type computer.

In order to execute tcACCESS as a Dynamic Link Library the global parameter `InProcess=1` must be specified in section `[CommunicationHandler]` of the tcACCESS file `tcACCESS.ini`.

Entry in file tcACCESS.ini:

```
[CommunicationHandler]
InProcess=1
```

Important notes:
Before changing parameter InProcess all active tcACCESS processes must be terminated (i.e. bmlch.exe, tcACCESS.exe, tcaLS.exe, tcACCESS ODBC aplications). It is generally recommended to restart the server computer after option InProcess has been changed.

When executing the Communication Handler as Dynamic Link Library no symbol and no Communication Handler Window is displayed.

7. Command Line Programs for the Communication between PC and Host

7.1 Supported Command Line Parameters

7.1.1 Overview

All available command line parameters are listed below:

PARAMETER	DESCRIPTION
/AI	Display of an input window for user name, password and new password
/AS	PC or ASA File Mode
/BT	Binary data transfer
/C3	3270 command
/CA	Start session
/CD	End session
/CE	Command line to be started
/CF	Process file with command line parameters
/CI	Establish host connection
/CM	User-defined error message
/CO	Comment for the command line to be started
/CP	VM/CP command to be processed
/CS	Character set
/CT	Stop host connection
/CX	Convert 3270 screen buffer into a ASCII file
/DA	Start date
/DR	Progress display for data transfer from host
/ED	Character for decimal point
/EK	Last key value or record number to be transferred
/EQ	Character for quote
/ES	Character for field separation
/FSF	Find slot without session
/FSS	Find slot with session

PARAMETER	DESCRIPTION
/FT	PC target file type
/GS	Get statistic from host
/HD	Start delay
/HF	Name of the host file
/HO	Start time
/HP	Host parameter
/HT	Host file type
/ID	User name
/JEBURS	JES burst
/JECLAS	JES class
/JECOPY	JES copies
/JEDENO	JES destination node
/JEDEUS	JES destination user
/JEFCBI	JES FCB name
/JEFLAC	JES flash count
/JEFLAS	JES flash name
/JEFORM	JES form
/JEHOLD	JES hold
/JEOUTP	JES output reference
/JEUCSI	JES UCS name
/JEWRIT	JES writer name
/KC	Keep CR/LF sequences
/LF	PC file name
/MC	Maximum length of character fields at SQL query
/MR	Maximum number of records to be transferred
/NA	Deactivation of **Cancel** key in status window for a data transfer
/NC	Add CR/LF per record
/ND	No display of host file name on status windows for a data transfer
/NK	Deactivate confirmation with key
/NT	Truncate trailing blanks

PARAMETER	DESCRIPTION
/NW	New password
/POBUIL	POWER building
/POCLAS	POWER class
/POCOPY	POWER copies
/PODEPT	POWER department
/PODISP	POWER disposition
/POFCBI	POWER FCB name
/POFORM	POWER form
/POJOBN	POWER job name
/PONSEP	POWER separator pages
/POORGN	POWER originator node
/POORGU	POWER originator user
/POPRGN	POWER programmer name
/POPRIO	POWER priority
/POROOM	POWER room
/POSUBS	POWER subsystem name
/POSYSI	POWER system ID
/POTGTN	POWER target node
/POTGTU	POWER target user
/POUINF	POWER user information
/POUSER	POWER user
/PW	Password
/RF	Create new target file
/RL	Length of a record
/RS	Record selection
/SC	Display status window
/SF	Process text file with SQL commands
/SI	Session ID for 3270 connection
/SK	First key value or record number to be transferred
/SL	Slot number to be used
/SQ	SQL command

PARAMETER	DESCRIPTION
/SS	FTB file (structure definition)
/ST	Structured data transfer
/SV	Name of the host server module
/UC	User-Catalogname of the VSAM file
/VL	VSAM file of the VSE-library
/VO	Volume name

The following table shows the relations between the parameters and the programs:

	FUNCTIONS											
	BMLGP			BML-UL	BML-DL	BMLSP			BML-CP	BML-SQ	BML 3270-	BML RM
PARAMETER	/ON	/OF	/TI			/PR	/RD	/PU				
/AI	O						O (1)					O
/AS						O		O				
/BT				O	O	O	O	O				
/C3											O	O
/CA											O	
/CD											O	
/CE			N									
/CF	O		O	O	O	O	O	O	O	O	O	O
/CI											O	
/CM											O	
/CO			O									
/CP									N			
/CS				O	O	O	O	O	O	O		
/CT											O	O
/CX											O	
/DA			O									
/DR					O					O		
/ED					O					O		
/EK					O							
/EQ					O					O		
/ES					O					O		
/FT				O	O					O		
/FSF	O											O
/FSS	O											O
/GS				O	O	O	O	O		O		
/HD			O									
/HF				N	N							O

	FUNCTIONS											
	BMLGP			BML-UL	BML-DL	BMLSP			BML-CP	BML-SQ	BML 3270-	BML RM
PARAMETER	/ON	/OF	/TI			/PR	/RD	/PU				
/AI	O						O (1)					O
/HO			O									
/HP				O	O							O
/HT				O								
/ID	O						O (1)					O
/JExxxx						O						
/KC				O								
/LF				N	N	N	N	N	N	O	O	
/MC										O		
/MR				O	O	O	O	O		O		
/NA				O	O	O	O	O	O	O		
/NC					O							
/ND					O							
/NK			O	O	O	O	O	O	O	O	O	O
/NT					O					O		
/NW	O											O
/OF												
/ON												
/POxxxx						O	O (2)	O				
/PW	O						O (1)					O
/RC												O
/RF				O	O							
/RL				O		O	O	O				
/RS					O							
/SC			O	O	O	O	O	O	O	O	O	O
/SF										O		
/SI	O										O	O
/SK				O	O	O	O	O				
/SL	O	O	O	O	O	O	O	O	O	O	O	O

	FUNCTIONS											
	BMLGP			BML-UL	BML-DL	BMLSP			BML-CP	BML-SQ	BML 3270-	BML RM
PARAMETER	/ON	/OF	/TI			/PR	/RD	/PU				
/AI	O						O (1)					O
/SQ										N		
/SS					O							
/ST					O							
/SV					O					O		
/UC												O
/VL												O
/VO					O							O

N = necessary

O = optional

(1) Only for an OS/390 Job Submit (see also /RD parameter)

(2) Only the first two parameters (job name and user) are used

7.1.2 /AI - Display of a User Logon Dialog

Open a dialog before the system establishes a connection. This dialog allows you to enter a user name and the related password.

You can also use the command line parameters /ID (page 348), /PW (page 350) and /NW (page 350) to define the values directly. The system then initializes the logon dialog with these entries.

The following example requires only the additional user input of the password since the user name is set to TEST and the new password field is empty.

Example:

```
BMLBAT /ON /ID TEST /NW /AI
```

7.1.3 /AS - PC or ASA Print File

Indicate that a PC print file contains an ASA special character at the beginning of each line. If you omit this parameter, the system automatically converts all *Line Feed* and *Form Feed* characters into ASA control characters.

7.1.4 /BT - Binary Data Transfer

Perform a binary data transfer. If you omit this parameter, depending on the direction of the transfer, the system translates data from ASCII to EBCDIC or from EBCDIC to ASCII.

7.1.5 /C3 - 3270 Commands

Process 3270 commands within the currently active host emulation.

Example:

```
BML3270 /C3 [KCESN] [AEN]
```

If you use the /LF parameter to define a PC file and the 3270 command [R] at the same time, tcACCESS saves a copy of the current 3270 screen buffer into this file. If you also use the /CX parameter (see page 339), the system creates an ASCII file with end-of-line control characters (CR/LF).
The system saves the current cursor position on the 3270 screen (in combination with operation [R]) or the number of the retrieved character string (with operation [W...]) in the ERRORLEVEL system variable. You can then process this variable via batch files.

7.1.6 /CA - Start Session

Start a mainframe session. If you are using a 3270 host emulation, the host connection must be established in advance (e.g. via parameter /CI, see page 338).

7.1.7 /CD - End Session

End an established host session. If you are using a 3270 host emulation for the connection with the mainframe, tcACCESS changes to 3270 input mode when it has processed the parameter. You can end the connection with the /CT parameter (page 339).

7.1.8 /CE - Command line to be started

Define a program (together with the required parameters) to be processed by the timer function via BMLGP /TI or BMLBAT /TI. This parameter must be at the last position of a command line.

Examples:

```
BMLBAT /TI /CE BMLDL /LF pc.dat /HF hostfile
BMLBAT /TI /CE BMLSHELL /C CALL mybatch.bat
```

7.1.9 /CF - File with Command Line to be processed

Pass required command-line parameters from an ASCII file (max 4000 bytes) to the desired program.

If you have defined the same parameter in both the text file and the command line, the latter has priority. Thus, entries in the parameter file may be overwritten.

Example:

If the bmlsql.txt file contains the following text:

```
/MR 10 /SQ SELECT * FROM SYSTEM.SYSCATALOG
```

The following call executes this SQL command and saves the resulting maximum of 20 lines in the result.dat file:

```
/LF result.dat /MR 20 /CF bmlsql.txt
```

7.1.10 /CI - Establish Host Connection

Establish a connection that allows you to pass 3270 commands to the system. If TCP/IP or APPC is used for the communication, establishing the communication ends with the user logon.
You must start a session using the /CA parameter (page 337) (in most cases with a valid user signon).
In order to overwrite the values predefined in the Connection configurations (page 190) dialog for a connection that is to be established, you can also use the /SI parameter (see page 355).

7.1.11 /CM - User defined Error Messages

Define the text for any error message to be displayed on the screen. The related error code is 30000000 Hex.
You can use the message to inform a user about a failed call within the logon and logoff batch files tcalogon.bat or tcalogof.bat.

Examples:

```
/CM Unknown host screen!
/CM Host target system is not active!
```

7.1.12 /CO - Description for a command line to be started

If you wish to perform a timer-controlled operation, use this parameter to define a description that displays until the specified time is reached.

Example:

```
/CO Delayed data transfer
```

7.1.13 /CP - VM/CP command to be processed

If a tcACCESS session with VM is established, use this parameter to submit a valid VM/CP command for the mainframe to process.

Example:

```
/CP QUERY COMMANDS
```

7.1.14 /CS - Define Character set

Define the character set to be used on the PC.

```
/CSD
```

DOS character set (OEM)

```
/CSW
```

Windows character set (ANSI)
If you do not use this parameter, the system uses the DOS character set by default.

7.1.15 /CT - Terminate Host Connection

Terminate the host connection if there is only one host connection and you have not started a session.

7.1.16 /CX - Save 3270 Screen buffer in ASCII File

If you use the 3270 command [R] in combination with the /LF parameter, the current content of the 3270 screen buffer saves into a file. If you also use /CX, the system saves the screen buffer file in ASCII format (with line feed control characters).

Example:

```
BML3270 /C3 [R] /LF 3270.dat /CX
```

7.1.17 /DA - Start Date

If you want the system to process commands on a timer-controlled basis, use this parameter to define a start date in dd-mm-yyyy format.
If you do not set a start date, tcACCESS uses only the start time.

Example:

```
/DA 25-05-1999
```

7.1.18 /DR - Percentage Display for Data Transfer from Host

Activate the percentage display for data-transfer operations from the host (file transfers and SQL accesses).

Note: Using this option may duplicate host accesses since this is the only method available for investigation of the resultant data volume.

Following host sources require double data access:

- VSAM KSDS, ESDS, RRDS
- TS Queues in combination with an SQL access
- MVS PDS/PS

This option requires at least tcACCESS version 2.2 on the host. In addition to the aforementioned file types, this parameter is also available for the following data sources (but without double access):

- ADABAS (if no WHERE condition is defined)
- Virtual disks

7.1.19 /ED - Character for Decimal point

Define the decimal point character. A period (**.**) is the default.

7.1.20 /EK - End Record or Number

Define a value for the last record to be transferred. You can specify either a number (beginning with 0) or a key to be translated into EBCDIC before it is passed to the host.
A detailed description of possible key values can be found under /SK - Start Record or Number (page 355).

7.1.21 /EQ - Quotes

Define the character used to delimit character string fields when transfer mode Separated fields is selected. A single quote (**'**) is the default.

7.1.22 /ES - Delimiter Character

If you have selected the transfer mode Separated fields, use this parameter to define a character to be inserted between the values of two fields. A comma (**,**) is the default.

7.1.23 /FSF xyz - Find a slot, which uses the defined configuration

tcACCESS tries to find a free slot which uses the defined configuration and has no active host session. If parameter "/SL" has been defined, it has a higher priority and prohibits the usage of "/FSF".

If no parameter is specified to define a slot ("/SL", "/FSF" oe "/FSS"), tcACCESS checks environment variable "DEFAULT_SLOT". This variable always exists, when a tcACCESS program has been started using the "tcACCESS Command Shell" (refer to The tcACCESS Command Shell BMLSHELL page *315)* and the variable contains a slot number. If this slot number does not exist, the slot number most recently used by the tcACCESS program for data communication between the PC and the mainframe will be used.

The number of the slot found will be returned as returncode to the caller. If no result could be determined, the value "-1" will be returned. With BMLSHELL, the returncode can be processed using the environment variable "ERRORLEVEL". See the following example:

```
bmlgp /fsf default
if %ERRORLEVEL% == -1 GOTO :ERROR
set DEFAULT_SLOT=%ERRORLEVEL%
```

7.1.24 /FSS xyz - Find a slot, which uses the defined configuration

tcACCESS tries to find a free slot which uses the defined configuration and has an allocated session. The session must not be active at the moment (e.g. active data transmission).

If all slots which might fit do have an active session, you can use the parameter "/ON". TcACCESS then tries to find a corresponding slot with no session and tries to establish a new connection.

In combination with option "Close inactive connection after.." /FSS can be used to do a "Connection Pooling" (refer to page 205).

If "/SL" has been specified, this parameter has a higher priority and prohibits the useag of "/FSS".

If no parameter specified to define a slot ("/SL", "/FSF" oe "/FSS"), tcACCESS checks environment variable "DEFAULT_SLOT". This variable always exists, when a tcACCESS program has been started using the "tcACCESS Command Shell" (refer to The tcACCESS Command Shell BMLSHELL page *315)* and the variable contains a slot number. If this slot number does not exist, the slot number most recently used by the tcACCESS program for data communication between the PC and the mainframe will be used.

The number of the slot found will be returned as returncode to the caller. If no result could be determined, the value "-1" will be returned. With BMLSHELL, the returncode can be processed using the environment variable "ERRORLEVEL". See the following example:

```
    bmlgp /fsf default
    if %ERRORLEVEL% == -1 GOTO :ERROR
set DEFAULT_SLOT=%ERRORLEVEL%
```

7.1.25 /FT - PC File Type

Define the target file type on the PC for structured data transfer and SQL queries.

The following values are allowed:

0 Standard ASCII file.
1 ASCII file with separated fields (e.g. CSV format (Comma Separated Variables)).
2 Database file in dBASE III format.

7.1.26 /GS - Generate Statistics

If you use this parameter, tcACCESS creates the bmlstat.txt file in the work directory (WORK) after a successful data transfer. This file contains statistical information which may, for example, be added to a log file in order to document automatically processed data transfers.

7.1.27 /HD - Start Delay

Define the start delay in the format hh:mm (max. delay: 23:59) for a timer-controlled transfer.
If you have set the parameters /DA for the start date and /HO for the start time, the system suspends them when this parameter is applied.

Example:

```
/HD 00:15
```

7.1.28 /HF - Host File name

Define the desired host file and also the directory entries to search for. You must observe the following syntax rules:

DL/I

PSB name with suffix (DLI) (see also parameter /SK, page 355)

Example:

```
PSB007(DLI)
```

MVS PDS/PS files

The data set name for a required member in brackets has to be positioned at the end.

Example:

```
QUAL1.QUAL2(MEMB)
```

Directory entries of MVS PDS/PS files

Prefix of the desired data sets or members followed by a ?

Examples:

```
QUAL1.?
QUAL1.QUAL2.(M?)
```

TD Queue

TD queue followed by (TD)

Example:

```
P860(TD)
```

TS Queue

Name of the TS queue followed by (TS)

Example:

```
MYQUEUE(TS)
```

Directory entries of TS queues

Prefix of the desired TS queues followed by ?(TS)

Example:

```
MYQ?(TS)
```

Virtual disks

Name of the virtual disk (HostFile.Partition for VSAM files, File name File type File mode for CMS files), a colon followed by the path specification for the desired file within the virtual disk

Example:

```
TCAVDI.DISK1:BAT\TEST.BAT
```

Directory entries for virtual disks

Name of the virtual disk (HostFile.Partition), a colon and the subsequent path to the directory which is to be considered for the search. The desired files and extensions can be specified with the ? place holder character.

Examples:

```
TCAVDI.DISK1:BAT\TEST.?
TCAVDI.DISK1:BAT\?.BAT
TCAVDI.DISK1:BAT\?.?
```

When directory entries of a virtual disk are transferred, the records are 44 bytes long.

VM/CMS

File name, file type and file mode (mini disk) separated from each other with blanks

Example:

```
TCAVDSK VDISK D
```

VSAM file

File name

Example:

```
MYFILE
```

Directory entries of VSAM files

Prefix of the desired files followed by a ?

Example:

```
MYF?
```

POWER Print Queue

LST(POWER) (or also the input of (POWER) only) followed by the parameters separated from each other with commas:

PARAMETER	DESCRIPTION
JN=xxx	JobName (Default = all)
CL=x	Class (Default = <blank>)
UI=xxx	User ID (Default = user who is currently signed on)
PW=xxx	Password
JR=xxx	JobNumber
JS=xxx	JobSuffix

Example:

```
(POWER),JN=JOB33,UI=US01,CL=Q
```

In order to transfer print control characters, use the parameters /HP CC=M (MCC format) or /HP CC=A (ASA format). When using an MCC format, the

system does not translate the first character of each record even if it converts the rest of the record between EBCDIC and ASCII.

In addition, you can submit a valid CTL command using the CD=xxx parameter.

Example:

```
(POWER),CD='D PUN',UI=US01
```

POWER Punch Queue

PUN(POWER) with the following parameters separated from each other by commas:

PARAMETER	DESCRIPTION
JN=xxx	JobName (Default = all)
CL=x	Class (Default = <blank>)
UI=xxx	User ID (default = currently signed-on user)
PW=xxx	Password
JR=xxx	JobNumber
JS=xxx	JobSuffix

Example:

```
PUN(POWER),JN=JOB33,UI=US01,CL=Q
```

POWER Reader Queue

RDR(POWER) with the following parameters separated from each other with commas:

PARAMETER	DESCRIPTION	
JN=xxx	=> JobName	(Default = all)
CL=x	=> Class	(Default = <blank>)
UI=xxx	=> User ID	(Default = user currently signed on)
PW=xxx	=> Password	
JR=xxx	=> JobNumber	
JS=xxx	=> JobSuffix	

Example:

```
RDR(POWER),JN=JOB33,UI=US01,CL=Q
```

Directory entries for POWER Queues

Definition of the desired queue (LST(POWER), PUN(POWER) or RDR(POWER)) followed by a JN= parameter which contains a place holder character * at the desired position.

Example:

```
PUN(POWER),JN=*QX
```

7.1.29 /HO - Start Time

If you wish to perform a timer-controlled data transfer, use this parameter to define the desired start time (format: hh:mm).

Optionally, you can also define a start date. Use the /DA parameter.

Example:

```
/HO 12:15
```

7.1.30 /HP - Host Server Module Parameter

This parameter allows you to pass specific information (separated by commas) to the host server module addressed.

The following entries are possible:

Transfer of host VSAM files to a PC

KL=n The key length of VSAM data sets is n bytes.

Example:

```
/HP KL=23
```

Transfer of PC files to host MVS PDS/PS files

PARAMETER	DESCRIPTION
DS=OLD	OLD disposition for write accesses (default)
DS=SHR	SHR disposition for write accesses
NW=Y	Create new MVS data set
NW=N	Do not create a new MVS data set (default)

The following parameters can only be used together with NW=Y:

PARAMETER	DESCRIPTION
UT=xxx	Unit xxx (UNIT)
VO=xxx	Disk name xxx (VOLSER)
RF=F	Records with a fixed length, unblocked
RF=FB	Records with a fixed length, blocked
RF=V	Records with a variable length, unblocked
RF=VB	Records with a variable length, blocked
LR=n	Logic record length is n bytes

PARAMETER	DESCRIPTION
BS=n	Block size is n bytes
DB=n	Number of directory blocks
PS=n	Primary space assignment
SS=n	Secondary space assignment
SP=B	Unit for space assignment is blocks
SP=C	Unit for space assignment is cylinder
SP=T	Unit for space assignment is tracks
AV=n	Size of a block for space assignment (only for SP=B)

Example:

```
/HP NW=Y,LR=80,BS=800,RF=FB,PS=1,SP=CYL,UT=SYSDA
```

Transfer of host MVS PDS/PS files to a PC

PARAMETER	DESCRIPTION
DS=OLD	OLD disposition for read accesses
DS=SHR	SHR disposition for read accesses (default)
VO=xxx	Disk name (VOLSER) for non-catalogued MVS PDS/PS files

Example:

```
/HP VO=T80S01
```

Transfer of host POWER files to a PC

PARAMETER	DESCRIPTION
CC=M	File contains MCC control characters
CC=A	File contains ASA control characters
IC=Y	Return direct commands

Example:

```
/HP CC=A
```

Transfer of host DL/I data to a PC

PARAMETER	DESCRIPTION
SL=n	Use fixed segment length n
SN=Y	Places the 8 characters long segment name in front of each record
KF=Y	Places the complete **DL/I** key in front of each record

FL=n	Records on the host are to be read with a fixed record length n instead of reading the individual fields by a CR, LF or CR/LF classification (default).

7.1.31 /HT - Host Data Type

When you wish to process a structured file transfer (/FT2), use this parameter to specify the host file type which is to be created. For this, the source must be a PC dBASE file.
The following values are possible:

- 10 Standard EBCDIC file, numeric fields are written as character strings
- 11 IXF file (in accordance with IBM specification)
- 12 Standard EBCDIC file, numeric fields are saved in zoned format
- 13 Standard EBCDIC file, numeric fields are saved in packed format

Example:

```
BMLBAT /UL /LF ADDRESS.DBF /FT2 /HF
TSO.IXF.DATA(ADDRESS) /HT11
```

7.1.32 /ID - User name

Define a user name for the start of a host session, up to 8 characters.
This value, along with the requested password, which must also be passed, identifies the user to the host. It is passed to each tcACCESS host server module used later.

Example:

```
/ID USER001
```

7.1.33 /JExxxx - The JES Parameters

The system initializes the parameters not provided in the command line using the file TCADEF.PPH located in the parameter directory (PARAM). You can create this file by selecting the menu entry Set as default in the PC => host transfer dialog.

The following are parameters for sending files to JES:

PARAMETER	DESCRIPTION
/JEBURS	JES Burst (may be Y or N)
/JECLAS	JES Class
/JECOPY	JES Copies (number)
/JEDENO	JES Destination Node
/JEDEUS	JES Destination User
/JEFCBI	JES FCB Name
/JEFLAC	JES Flash Count

PARAMETER	DESCRIPTION
/JEFLAS	JES Flash Name
/JEFORM	JES Form
/JEHOLD	JES Hold (may be Y or N)
/JEOUTP	JES Output Reference (e.g. STEP1)
/JEUCSI	JES UCS Name
/JEWRIT	JES Writer Name

7.1.34 /KC - Keep CR/LF

If you use this parameter for a text transfer to a host, no line-feed sequences contained therein (CR/LF) are removed. Instead, the system transfers and translates them like any other character. This function is mainly useful for SWIFT files.

7.1.35 /LF - PC File name

Define the source or target file on the PC.

Example:

```
/LF C:\DATA\MYFILE.DAT
```

7.1.36 /MC - Maximum Length of Character String Fields

Define how many characters of character string fields should be sent to PC when transferring data from the host.

Example:

```
/MC 72
```

7.1.37 /MR - Maximum Number of Records

Define a maximum number of records to be transferred. By default the system transfers all records.

7.1.38 /NA - Cancel Button Deactivation

Block the Cancel button in the status window (using the /SC parameter) of a data transfer. Thus, the user cannot terminate the transfer before it is completed.

7.1.39 /NC - Add no CR/LF Sequence

When data is transferred into a PC text file, a CR/LF sequence is added to each record by default. Use the /NC parameter to suppress the addition of the CR/LF sequence.

7.1.40 /ND - Special Screen Output Suppression

In order to prevent the display of the used host file in the status window (using the /SC parameter), the value H must be defined behind this parameter (i.e. /NDH) when transferring data.

7.1.41 /NK - Data transfer without User Confirmation Request

If you want the status window (parameter /SC) to close automatically when a data transfer has completed successfully, use this parameter. Otherwise, tcACCESS expects the user to press the appropriate button.

7.1.42 /NT – No Truncation of Trailing Blanks

When host data is written to a PC text file, the system removes blanks at the end of a record by default. Also, the blanks at the end of each character string are removed when a dBASE III file is written on a PC. If you do not want blank truncation, use the /NT parameter.

7.1.43 /NW - New Password

If a new password is to be used on the host, use this parameter (followed by an password, up to 8 characters) in addition to the parameters for the user name (/ID).

Example:

```
/NW NEWPASS
```

7.1.44 /OF - Terminate session

This function terminates an active session with the host. If exist, batch-file "TCALOGOF.BAT" will be executed to close host applications (if necessary). If the batch does not exist, the logoff-sequence defined as part of the configuration will be executed and the host connection will be terminated.

Example: BLMGP /SL 1 /OF

7.1.45 /ON - Start session

In case the PC Communication Handler has not yet been started or not connection to the host exists these function will now be performed. If the connection has been established using a 3270-emulation, the logon-sequnece defined as part of the configuration will be executed.
The slot selection is performed using parameter "/SL", "/FSF" or "/FSS".

Example: BMLRM /SL 5 /ON /HF TEST(TS)

7.1.46 /POxxxx - The POWER Parameters

The system initializes the values not given in the command line by using the file TCADEF.PPH in the tcACCESS parameter directory (PARAM). You can create this file by selecting the menu entry Set as default in the PC-to-host transfer dialog.

Use the following parameters for sending files to POWER:

PARAMETER	DESCRIPTION
/POBUIL	POWER Building
/POCLAS	POWER Class
/POCOPY	POWER Copies (number)
/PODEPT	POWER Department
/PODISP	POWER Disposition
/POFCBI	POWER FCB Name
/POFORM	POWER Form
/POJOBN	POWER Job Name
/PONSEP	POWER Separator Pages (number)
/POORGN	POWER Originator Node
/POORGU	POWER Originator User
/POPRGN	POWER Programmer Name
/POPRIO	POWER Priority
/POROOM	POWER Room
/POSUBS	POWER Subsystem Name
/POSYSI	POWER System ID
/POTGTN	POWER Target Node
/POTGTU	POWER Target User
/POUINF	POWER User Information
/POUSER	POWER User

7.1.47 /PW - Password

When a session is started, use this parameter to transmit the password belonging to the specified user name (/ID). It may be up to 8 characters long.

Example:

```
/PW MYPASS
```

7.1.48 RC xyz

Definition of an alternate file for error messages.

xyz => fully qualified path to error file
In case that you do not want that the error messages that can occur during a datatransfer will be written into file "tcaretXXX.txt" ("XXX" equals to the slot

number) in directory "WORK", you can use this parameter to specify an alternate file. This can be used if multiple calls are being executed simultaneously (i.e. by the tcACCESS Listener) and possible error files must be processed individually
.

Example: BMLRM /SL 10 /ON /HF TEST(TS) /RC C:\temp\error.txt

7.1.49 /RF - Restore Target file before Writing

Restore (delete) the target file, if possible, before writing. By default the transmitted data is appended to the end of an existing file.

7.1.50 /RL - PC File Record Length

If a fixed record length is used for the source file on the PC, use this parameter to define the desired length.

7.1.51 /RS - Record Selection

Transfer (together with the /EK and /SK parameters) selected records from the host to a PC. You can load a description of the addressed fields using the parameters /ST or /SS.

Syntax format of conditions:

```
/RS [NOT] field [NOT] Operator value [AND ¦ OR] ...
```

Field

In most cases this is a field of the description file which is admitted for queries.

A field may also be entered in the following form, when a condition refers to a undescribed field:

Field type (position, length, fractions)

Field type:

- C Character
- B Binary
- P Packed
- Z Zoned
- H Halfword (2 bytes with sign)
- F Fullword (4 bytes with sign)
-

Position:

- Start position of the field relative to the record begin (first byte has position 0)
-

Length:

- Character, binary Number of characters
- Packed, Zoned Number of decimals
- Halfword Always 2 bytes
- Fullword Always 4 bytes
-

Fractions:

- Number of decimal places behind the comma (only for field types halfword, fullword, packed and zoned).
-

Example:

```
/RS C(0,20) = 'Miller' OR C(0,20) = 'Meyer' AND
P(20,5,0) = 12345
```

Operator:

The following operators are available:

OPERATOR	DESCRIPTION
<	Less than
<=	Less than or equal to
=	Equal to
<>	Not equal to
>=	Greater than or equal to
>	Greater than
LIKE	Equivalent to (of character strings); may use wildcards

Value

For character and binary fields, you must enclose the value in single quotes ('). Hexadecimal comparison values can be specified in the form H'xxx'. Numeric values with a sign and a decimal point or comma may be entered as follows:

If operator LIKE or NOT LIKE is selected, the value may contain the following wild card characters:

CHARACTER	DESCRIPTION
_ (underscore)	Represents an optional individual character
%	Represents an optionl character string

Example:

```
%A_CD
```

selects: ABCD YZABCD ABCAXCD
but does not select: AB XYZCD AXYCD

AND | OR

You can control the sequence of processing using parentheses. If you do not use parentheses, tcACCESS processes AND connections first and OR connections afterward.

Examples:

Selects all 'Miller' and 'Smith' records in AREA NY.

```
/RS (NAME = 'Miller' OR NAME = 'Smith') AND AREA = NY
```

Input for the selection of all 'Miller' records and additionally all 'Smith' records in AREA NY:

```
/RS NAME = 'Miller' OR NAME = 'Smith' AND AREA= NY
```

7.1.52 /SC - Status window

Display a status window during a data transfer process. The window displays until the user cancels the transfer or until the transfer has been successfully completed.

7.1.53 /SF - File with SQL Commands

Define a text file containing SQL commands. This allows you to transfer several SQL commands to the server at once. A description of the SQL command syntax can be found under the description of the /SQ parameter.

7.1.54 /SI - Connection Parameter Performation

If you are using a 3270 emulation for connection with the host, use this parameter to determine a session ID different from the one defined in the tcACCESS main program (see Connection configurations, page 190). You can then use this ID to establish the host connection.

Example:

```
/SI C
```

If you are using a TCP/IP connection, use this parameter to overwrite the defined values with the TCP/IP address and/or port. You must use the following syntax:

```
/SI [HostName|HostAddress] [,Port]
```

Examples:

```
/SI 157.31.3.40,23
/SI pmvs05
/SI ,23
```

7.1.55 /SK - Start Record or Number

Define the first record to transmit. The syntax depends on the host file type used:

- sequential file (VSAM ESDS, VM/CMS, TS Queue, MVS PDS/PS)
  The record is determined by its record number (first record starts with number 0)

-
- File with key (VSAM KSDS, DL/I)
  The record is determined by its key.
-

Key input for VSAM KSDS:

You can enter text directly, and the system automatically translates it into EBCDIC. For numeric key values see Definition of numeric key values in the following section.

Specification of numeric key values:

You can enter numeric data as follows:

```
%%CData%%
```

C represents the type of field and Data its contents.

The following field types are supported:

FIELD TYPE	DESCRIPTION
Hxxx	Hexadecimal number (up to 14 characters)
Px1,x2	Packed, x1 = number of positions, x2 = value (up to 18 characters)
Ixxx	Integer (fullword)
Sxxx	Small Integer (halfword)

Examples:

```
/SK %%P4,-123%%
```

(4 digit packed field; equivalent to the hexadecimal value 00123D)

```
/SK %%I033%%
```

(Fullword; equivalent to the hexadecimal value 00000021)

```
/SK %%H321%%
```

(2 bytes in hex format; equivalent to the hexadecimal value 0321)

7.1.56 /SL - Specification of a slot number

This parameter defines the slot number to be used.

Example:

```
/SL  5
```

7.1.57 /SQ - SQL Command

The system processes the SQL command entered after this parameter on the host and returns the result to the PC. The /SQ parameter must always be at the end of a command line.

Example:

```
/LF SQL.DAT  /SQ SELECT * FROM SYSTEM.SYSCATALOG
```

Depending on the selected database type (or tcACCESS host server module with /SV parameter) different SQL commands may be used (by default DB2 is addressed in an OS/390 environment and SQL/DS in VSE and VM system environments):

DB2

(Host server module: TCASSQLQ)

It is possible to specify several SQL commands at once, separated from each other with a semicolon. They are processed as LUW, i.e., the system either executes all of them or none if an error occurs.

The last SQL command only can be a query command (SELECT ...). The system ignores all SQL commands after the first query command.

DB2 supports the following entries:

- Standard commands such as SELECT ... or UPDATE ...
- Comments identified by two hyphens at the beginning (e.g.: -- This is a comment;)
- COMMIT
  Confirms the current LUW and activates the autocommit mode.
- ROLLBACK
  Terminates the current LUW, reverses data modifications of this LUW and activates the autocommit mode.
- NOCOMMIT
  Deactivates the autocommit mode. The client application must then explicitly perform a ROLLBACK or COMMIT in a consecutive SQL command in order to close the LUW. tcACCESS automatically executes a ROLLBACK, when either a non-SQL function is started or a host session is ended normally or abnormally.
-

Note: SQL commands following a NOCOMMIT command are processed with an ISOLATION LEVEL(RR) instead of the standard ISOLATION LEVEL(CS), i.e. data choosen with a SELECT command cannot be modified by the user before the LUW has completed.

Stored Procedures

DB2 Stored Procedures are called as follows:

```
CALL procedure (parm1, parm2, ...)
```

procedure represents the name of the procedure as it is defined in the DB2 system table SYSIBM.SYSPROCEDURES.PROCEDURE.

The further options **parm1**, **parm2**, etc. are equivalent to the parameters defined in the PARMLIST field of the called procedure. The CALL command must contain only those parameters defined as IN or INOUT. OUT and INOUT parameter values are passed to the client in the same way as the

result of a query. If there are no parameters of this type defined, then the system treats the CALL command like a non-query command.

SQL/DS (or DB2 for VSE or DB2 for VM)
(Host server module: TCASSQLQ)

You can specify several SQL commands separated from each other with a semicolon at once. They are processed as LUW, i.e. the system either executes all of them or none if an error occurs.

The last SQL command only may be a query command (SELECT ...). All SQL commands after the first query command are ignored.

The following input is supported under SQL/DS:

- Standard commands like SELECT ... or UPDATE ...
  Comments identified by two hyphens at the beginning (e.g.: -- This is a comment;)
-
- COMMIT
  Confirms the current LUW and activates the autocommit mode.
-
- ROLLBACK
  Terminates the current LUW, reverses data modifications of this LUW and activates the autocommit mode.
-
- NOCOMMIT
  Deactivates the autocommit mode. The client application must then explicitly process a ROLLBACK or COMMIT in a following SQL command in order to close the LUW. tcACCESS executes automatically a ROLLBACK when either a non-SQL function is started or a host session is ended normally or abnormally.
-

SQL-Engine
(Host server module: TCASTCAQ)

The SQL-Engine allows you to perform queries on non-SQL databases by using standard SQL commands. Supported are VSAM, DL/I, TS queues, MVS files, DATACOM/DB, ADABAS and others. Joins can used to retrieve data from different data sources. Views can be helpful to provide an exactly defined window on existing data for users.

Following functions are available:

- SELECT … FROM … WHERE …
  Retreival of data records (inclusive GROUP BY, HAVING, ORDER BY)
- INSERT … INTO … VALUES …
  Insert data records (optionally with subselect)
- UPDATE …
  Update data records (optionally with subselect)

- CALL
  Call stored procedure
- DELETE … FROM … WHERE …
  Delete data records
- COMMIT
  Confirm all operations since the beginning of this transaction
- ROLLBACK
  Undo all operations since the beginning of the transaction
- NOCOMMIT
  Deactivate the automatic commit
- **"EXECUTE"**
  Execution of a Precompiled Command

In addition, the following functions are supported within expressions:

- Single functions
  BIN(), DATE(), DAYS(), DECIMAL(), DECODE(), HEX(), SUBSTR(), STR(), TIME(), TIMESTAMP(), USER(), CURRENT_ TIMESTAMP()
- Group functions
  AVG(), COUNT(), MAX(), MIN(), SUM()
- Operators
  +, -, *, /, ||
- Condition operators
  <, >, =, >=, <=, <>, BETWEEN, LIKE, IN, IS NULL
- Logic operators
  AND, OR

For detailed information on the SQL-Engine see the *tcACCESS Host Server* manual.

IXF File

IXF-formatted data can be transferred from an MVS PDS/PS in the same way as from a database. For this, apply the following SQL command:

```
IXF=dsn
```

dsn represents a valid name of an MVS PDS/PS file.

7.1.58 /SS - Loading a Structure Definition

If a structured transfer should be avoided in order to load a structure definition (FTB file) for the usage of the /RS parameter, use this parameter instead of an /ST can be used.

Example:

```
/SS C:\FILE1.FTB
```

7.1.59 /ST - Structured Data transfer

Different from the /SS parameter, this parameter loads a structure definition (FTB file), but also performs a structured data transfer simultaneously.

Example:

```
/ST C:\FILE1.FTB
```

7.1.60 /SV - Host Server Module

Define a host server module as target for processing. This function is helpful if, for example, any other database system than DB2 or SQL/DS is accessed for an SQL query.

Example:

```
/SV TCASTCAQ
```

It is also possible to address individually programmed host server modules. They must generate an output which is compatible with the Browse function of the tcACCESS GPS modules (General Purpose Server). As an alternative method data may be pre-processed by an individual module (e.g. copying data from a database into a TS queue) and then be passed to a GPS module.

7.1.61 "/UC" – VSAM-Catalog

Use this parameter to specify the catalog of a VSAM file (i.e for file-type VSE-Library(Catalog))

Example: /UC USER.CATALOG.SYSWK2

7.1.62 "/VL" – VSAM-file of a VSE-Library

Use this parameter to specify the VSAM file of a VSE library.

7.1.63 "/VO" - VOLUME Name

Use this parameter to specify the volume name for the file that should be downloaded or deleted.

Example: /VO SYSWK5

7.2 Processing tcACCESS Command Line Functions with BMLBAT

BMLBAT allows you to start all command line functions available for the communication between PC and host. This program verifies the defined parameters and calls the appropriate supplementary tcACCESS program.

The following elementary functions are supported:

FUNCTION	DESCRIPTION
/32	3270 commands (Supplementary program: BML3270.EXE)

FUNCTION	DESCRIPTION
/CP	Execute a VM/CP command (Supplementary program: BMLCP.EXE)
/DL	Transfer data from host to a PC (Supplementary program: BMLDL.EXE)
/ON	Start session (Supplementary program: BMLGP.EXE)
/OF	End session (Supplementary program: BMLGP.EXE)
/TI	tcACCESS timer function (Supplementary program: BMLGP.EXE)
/PR	Send print file to host print queue (Supplementary program: BMLSP.EXE)
/RD	Send job file to host reader queue (Supplementary program: BMLSP.EXE)
/PU	Send data to host punch queue (Supplementary program: BMLSP.EXE)
/SQ	Execute a SQL command on the host (Supplementary program: BMLSQ.EXE)
/UL	Transfer data from PC to Host (Supplementary program: BMLUL.EXE)

The advantage of calling utility programs via the BMLBAT program is that this program waits until the respective program has ended. This is especially important in combination with batch files because otherwise no return codes can be verified, leaving no way to perform any program branches based on the return codes' values.

7.3 Sending 3270 Commands with BML3270

Send 3270 command sequences to the mainframe via an existing host connection.
Use batch files (extension .BAT) to create intelligent logon and logoff scripts which can use this command to access a 3270 emulation.

The following example sends the key input test to a connected 3270 emulation:

```
BMLBAT /32 /C3 [Ktest]
```

7.4 Executing VM/CP Commands on the Host with BMLCP

If a VM session is established with the tcACCESS assistant program, BMLCP allows you to send a valid CP command to the mainframe. You can save the result as text in a PC file.

Use the following command line to send the CP command QUERY COMMANDS to VM and save the result in the file OUTPUT.TXT on the PC:

```
BMLCP /CP QUERY COMMANDS /LF OUTPUT.TXT
```

7.5 Sending Host Data to a PC with BMLDL

Download the following data types from the mainframe:

- VSAM files
- MVS PDS/PS
- DL/I
- TS queues
- Files in virtual disks
- POWER queue entries
-

The transfer may either be performed as a binary transfer with an EBCDIC to ASCII translation or as a structured transfer.

The following example adds the content of the TS queue MYQUEUE to the end of the local file PC.DAT. A status window displays (/SC) and closes automatically at the end (/NK).

```
BMLDL /HF MYQUEUE(TS) /LF PC.DAT /RF /SC /NK
```

7.6 Host Logon and Logoff with BMLGP

Activate and deactivate a host session. The following parameters are available for this purpose:

/ON - Start session

If the PC Communication Handler is not started yet or a host connection does not yet exist, establish it.

Parameter "/SL" (page 356), "/FSF" (page 340) or "/FSS" (page 340) specify which slot should be used.

If you use a 3270 emulation for the connection, the system processes the logon sequence defined in the tcACCESS main program (see Logon/Logoff, page 193). If the TCALOGON.BAT batch file exists, the system executes it for a host logon (see also example file TCALOGON.SAM).

Example:

```
BLMGP /FSF /ON /ID user1 /PW passw1
```

or

```
BMLGP /ON /AI
```

/OF - End session

This function ends an active host session. If the TCALOGOF.BAT batch file exists, the system executes it in order to leave a host application. If this is not the case, the system processes the logoff sequence defined in the tcACCESS main program (see "Edit 3270 command" Dialog, page 195).

When the system completes these actions, the connection ends.

Example:

```
BLMGP /SL 1 /OF
```

7.7 Sending Data to a Host Print System with BMLSP

In order to pass a file to a JES or POWER queue, use the command line program BMLSP.EXE. It performs the following basic functions:

7.7.1.1 Pass a Print File to a Host Spool System (/PR)

tcACCESS allows you to transfer pure ASCII files or also files with ASA control characters at the beginning of each line to a print system on a mainframe.

Example:

```
BMLSP /PR /LF PRINT.DAT
```

7.7.1.2 Pass a Job to a Host Spool System (/RD)

You can pass a PC file containing a job description to a host spool system for execution.
If a +INC FILE.EXT is at the beginning of a PC file the system adds the respective file FILE.EXT to the job. These instructions may go 10 file levels down. If instead of +INC the +INCP key word is found, the added ASCII file is converted into ASA control characters for the host.
Thus, a batch job may be executed on an OS/390 host, which then prints a data file in accordance with the job's specifications.

Example:

```
BMLSP /RD /LF JOB.DAT
```

7.7.1.3 Executing OS/390 Jobs

In order to process jobs with variable values for the USER= and PASSWORD= parameters, tcACCESS looks for the following character strings and replaces them as described below:

+INCPCUI

If tcACCESS finds this character string somewhere in the job file, it replaces the string with the user identification. When the command line parameter

/ID is defined, the respective value is used. Otherwise, a window displays for the input of the user name.

+INCPCPW

If tcACCESS finds this character string somewhere in the job file, it replaces the string with the user password. If you use the command line parameter /PW, tcACCESS uses the associated value. Otherwise, a window displays so the user can input the password.

+INCHSUI

tcACCESS sends this character string to the host server module without any modification. If the system then finds an identical character string found somewhere in the job, it replaces the string with the name of the currently signed on user.

The host server module can be customized differently for executing jobs (TCASRDRQ). It allows the following requests:

- each job
- only jobs with USER= in the job card
- only jobs with USER= and PASSWORD= in the job card
- only jobs with USER=+INCHSUI in the job card

7.7.1.4 Transferring Data to a Host Punch Queue (/PU)

You can also transfer a PC file to the punch queue of a host spooling system (see also Pass a Job to a Host Spool System (/RD) above).

Example:

```
BMLSP /PU /LF JOB.DAT
```

7.8 Processing a SQL Query on the Host with BMLSQ

Execute a host SQL query. The output of a SELECT command can be saved in a PC file. The /SQ parameter must come last within a command line.

Example:

```
BMLSQ /LF SQL.OUT /SQ Select * from SYSIBM.SYSTABLES
```

7.9 Sending Data to a Mainframe with BMLUL

Files from a PC can be transmitted into a host file. Supported target formats are VSAM files, MVS PDS/PS, TS queues, TD queues and others. The transfer may either be performed binary or with an ASCII to EBCDIC translation.

Example:

```
BMLUL /LF PC.DAT /HF USER.TEST.SOURCE(PCDAT)
```

7.10 Converting Data in MCC Format with BMLCC

The DOS program BMLCC allows you to convert MCC format print files into ASCII files. Each file record in MCC format starts with a control character.

You can control the program using command line parameters:

- /IF name Name of the input file in MCC format.
- /OF name Name of the output file to be created.
- /CF name Name of the conversion file. (Description follows in the section below.)
- /MR n Maximum number of records to be written; if this parameter is not
- transmitted, all are converted.
-

The conversion file specified in parameter /CF must be an ASCII file with the following elements:

*** (at beginning of a line)**

Identifier for a comment line.

IGNORE_FIRST_NEWPAGE

If a form feed character is the first character of an output file, then it is suppressed.

TRANSLATE=x1;x2 comment

The MCC control character with the hexadecimal value x1 is translated into the MCC control charcater with the hexadecimal value x2 before the conversion. Each MCC value that is not processed with such a line is converted into 0, and is omitted for this reason.

The following is an example of a conversion file:

```
* Example Conversion File for BMLCC.EXE
*
* Options:
IGNORE_FIRST_NEWPAGE
*
TRANSLATE=bd;00    Ignore hex 'BD'
TRANSLATE=01;00    Ignore hex '01'
TRANSLATE=8b;8b    'Channel 1' becomes 'FormFeed' (new page)
TRANSLATE=09;09    Write line with CR/LF at the end
*                  (line feed)
TRANSLATE=19;19    Write line with three CR/LF at the end
*                  (several line feeds)
*
* All other input is converted ino hex '00' and will
```

* therefore by ignored

The following table shows how the system converts the existing MCC control characters into ASCII control characters after they have been processed with the conversion file:

7.11 BMLEXIST: Verifying the Existence of a Host File

The BMLEXIST.EXE command line program allows you to verify whether or not a certain file exists.
The following file types are supported:

- VSAM file
- TS Queue
- MVS PDS/PS

BMLEXIST is called with an /HF parameter.

Example:
BMLEXIST /HF QUAL1.QUAL2.SOURCE(TEST)

The ERRORLEVEL variable contains the result. It may assume the following values:

result	description
30	Communication error
20	Error of the host server module
7	User is not authorized for the operation
6	Searched file is not an MVS PS but an MVS PDS
5	Searched file is not an MVS PDS but an MVS PS
4	CICS dataset is not opened
3	MVS dataset is already in use
2	MVS PDS member not found
1	Dataset or queue not found
0	Dataset or queue found

In case of a communication or a host server module error BMLEXIST issues an appropriate message which may also be passed to a file by using the '>' operator.

7.12 Executing BMLSTAT for Connection Status Determination

The BMLSTAT.EXE program returns the current connection status of tcACCESS in the ERRORLEVEL system variable (see also the description of the Shell command IF on page 317).

The following return codes are available:

RESULT	DESCRIPTION
10	Unknown error.
3	A host session is started.
2	A host connection does exist (3270 mode).
1	The PC Communication Handler is loaded, but there is no host connection.
0	The PC Communication Handler is not loaded.

(See also example batch file TCASTAT.BAT)

7.13 "BMLRM": Delete a host file

Use BMLRM to delete host files. Based upon the type of file either the content of the file will be deleted by BMLRM (VSAM, PDS/PS) or the entire file will be physically deleted. The name of the file must be specified using the /HF parameter. The snytax of the host filename corresponds to the syntax used in the tcACCESS front-end program (i.e. FILE(TS) for a TS-Queue, DEMO.FILE(SAM) for a SAM-file etc.).

The following filetypes are supported:

- **SAM**
  Parameter **/VO** (Volume name) must be defined.

  Note:
  - Only the extents of the SAM-file on the specified volume will be deleted
  - Additional extents on another volume must be deleted with another execution of BMLRM and another /VO parameter.

  Example:

  /BMLRM /SL 1 /VO DOSRES /HF TEST.FILE(SAM) /SC /NK
- **TS-Queue**
- **logical files on Virtual Disk**

  Note:
  - For the specification of the filename you must use a Backslash (\) as directory separator (i.e. TCAVDI.SYSDSK:\TABLES\DEMO\ARTICLE.FTB)

- **VSE-Library**
- **VSE-Library (Catalog)**

Parameter /VL (VSAM-file for VSE-library) must be defined for type VSE-Library(Catalog).

- **VSAM-file (FCT)**
- **MVS PDS Member**
- **MVS PDS file**
  All members of the specified PDS will be deleted. The PDS file itself will not be deleted.
- **MVS PS**
- **POWER (Print-Queue, Punch-Queue, Reader-Queue)**

  Example:

  /BMLRM /SL 1 /SC /NK /HF LST(POWER),JN=CICS2,JR=01253,CL=A,ND=,UI=USR1,TU=USR1

8. The tcACCESS ODBC driver

The tcACCESS ODBC driver allows you to handle database queries on a mainframe via the standard Windows database interface ODBC. The TCAODBC(U).DLL file needed for this purpose is located in the base directory of tcACCESS.

Starting with Version 8.0 two ODBC driver are distributed. The ANSI-driver uses the EBCDIC/ASCII translation of the tcACCESS version up to and inclusive version 7. The UNICODE driver uses the "Global Language Pack" on the mainframe, if ainstalled.

The following host data resources are supported:

- **tcACCESS SQL-Engine**
  Allows you to access the following data sources:
  - IDMS
  - DATACOM
  - ADABAS
  - Non-SQL data sources, such as VSAM files, MVS PDS/PS, DLI, POWER, CICS TS and TD queues
- **DB2**
- **SQL/DS**
-

Details About ODBC Client Programs

Following are important details about ODBC client programs:

Microsoft Query

When you use a SELECT instruction, Microsoft Query always sorts table names in ascending alphabetical. You may prefer not to access data using the tcACCESS SQL-Engine since the sequence of the addressed tables will affect the outcome of your queries.

To avoid this outcome, you can generate a View that corresponds to the query by using the tcACCESS SQL-Engine. You can use this view to transfer the data.

8.1 The "tcACCESS ODBC Setup" Dialog

After a successful installation, this dialog allows you to customize a tcACCESS ODBC driver DSN. The process to start this dialong depends on the operating system.

Windows 3.x / NT 3.51

- Program group "Main"

- Icon "Control panel"
- Entry "ODBC (32Bit)"
- Data source "tcACCESS"
-

Windows 9x / ME / NT 4.0

- Start menu item "Settings"
- Sub-item "Control panel"
- Entry "ODBC (32Bit)"
- Data source "tcACCESS"
-

Windows 2000 / XP

- Start menu item "Settings"
- Sub-item "Control panel"
- Sub-item "Maintenance"
- Entry "Data sources (ODBC)"
- Data source "tcACCESS"

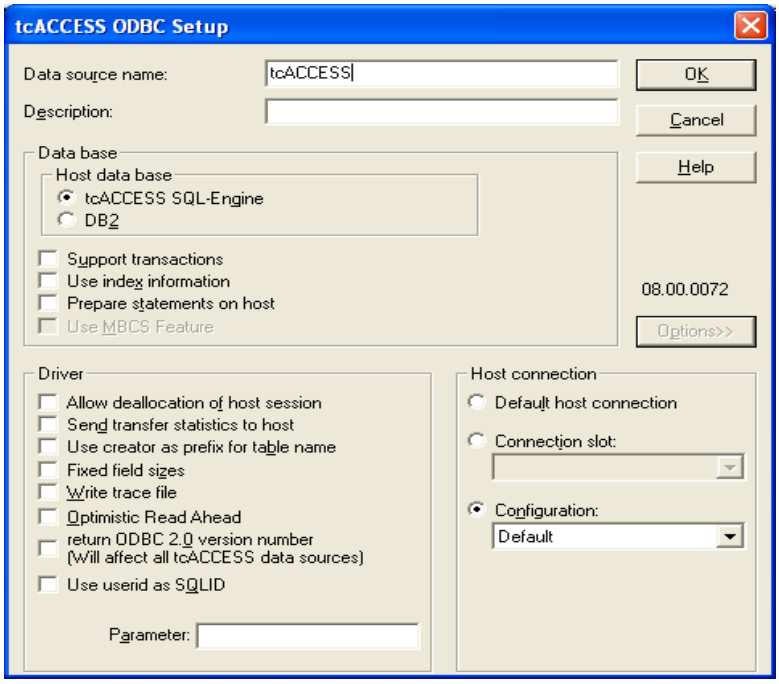


Fig. 215: tcACCESS ODBC Settings

The dialog components are described below:

Data source name

This name appears as selectable data source (DSN) in the Windows application and performs a query via ODBC (e.g., tcACCESS DB2).

Description

You may use this field to enter a description for the data source.

ODBC parameter

Key: *Description*

Host database

Use this field to select the preferred database type. The following types are available:

- **tcACCESS SQL-Engine**
  Allows you to access the following data sources:
  - IDMS
  - DATACOM
  - ADABAS
  - Non-SQL data sources, e.g. VSAM files, MVS PDS/PS, DLI, POWER
- **DB2**
  - Under OS/390
- **SQL/DS** (or "DB2 for VSE" or "DB2 for VM")
  - Under VSE or VM

ODBC parameter

Key: *Type*

Default value: *TCA/SQLX*

Possible settings:

TCA/SQLX (tcACCESS SQL-Engine), DB2 (DB2 or SQL/DS)

Support transactions

This option is available for the host database types DB2 ,SQL/DS and tcACCESS SQL-Engine. For tcACCESS SQL-Engine databases the option is available only if logical transactions are supported by the database itself. For example: DL/I databases support logical transations. The same applies to VSAM files with CICS Journalling/Recovery. On the contrary, PS/PDS, SAM files or VSAM Batch files do not support logical transactions. If it is selected, the ODBC support is activated by logic transactions. It is the responsibility of the calling Windows application to use this support appropriately.

If the support option is inactive, there is no ODBC support available for logic transactions. However, you can still submit the following standard ODBC SQL commands, for example, by using PASS THROUGH QUERIES in Microsoft Access:

COMMAND	DESCRIPTION
NOCOMMIT;	Ends the automatic COMMIT
COMMIT;	Closes a transaction

COMMAND	DESCRIPTION
ROLLBACK;	Undo a transaction

ODBC parameter

Key: *Transactions*

Default value: *No*

Possible settings: *Yes, No*

Use index information

When this option is active, tcACCESS returns index information to the ODBC client.

By using this information, the ODBC client application can perform INSERT, UPDATE, and DELETE operations. Therefore, it is necessary that the current user have the appropriate mainframe authorizations.

When host data and an ODBC client are incompatible, problems may occur. For example, the TIMESTAMP fields in DB2 contain six decimal positions for a fraction of split second. This is supported by Microsoft Query, but not by Microsoft Access. If a TIMESTAMP is used as primary key, a problem occurs with Microsoft Access when the fraction of a second value is unequal zero.

ODBC parameter

Key: *Index*

Default value: *No*

Possible settings: *Yes, No*

Use creator as prefix for table name

If you select this option, each table name returned to the calling program is preceded by the table creator and a dot (e.g., CREATOR.TABLE). In this case the creator is not returned separately (table without a creator).

This function is needed for PC applications that do not request creator names (e.g., Microfocus NetExpress) since this information is necessary for accessing most host database tables.

ODBC parameter

Key: *UseCreatorAsPrefix*

Default value: *No*

Possible settings: *Yes, No*

Activation of 'Optimistic Read Ahead'

When using this switch, the 'Optimistic Read Ahead' feature will be activated in the ODBC-driver.

'Optimistic Read Ahead' results in an immediate execution of the next request for data to the mainframe when the block of data has been received. The request will be processed by the tcACCESS mainframe component at the same time when the resultset is being passed to the application by the ODBC driver. This will result in a better performance when processing a large number of rows from the same table.

ODBC parameter

Key: *OptimisticReadAhead*

Default value: *No*

Possible settings: *Yes, No*

Fixed field sizes

If this option is set, trailing blanks of a field with type "string" are not removed.

ODBC parameter

Key: *FixedFieldSize*

Default value: *No*

Possible settings: *Yes, No*

Allow deallocation of host session

If this option is set, a btcACCESS host session established by an ODBC driver is automatically terminated during a "SQLFreeConnect" (e.g. at termination of MS ACCESS). Sessions already established are not concerned, even if this option is set.

If a user establishes a session, a free slot is being used for the desired configuration. This connection then will be used. If a second connection is needed, the first connection will be used, if this connection does not currently support an active request. Otherwise a new connection will be established. This leads into a Pool of multiple connections. If a connection is inactive for a defined time interval ("Close inactive connection after"), it will be automatically closed.

ODBC parameter

Key: *AllowDeallocSession*

Default value: *No*

Possible settings: *Yes, No*

Send transfer statistic to host

When this option is set, a message is sent to the mainframe at the end of each data transfer, indicating the statistics about the transfer.

ODBC parameter

Key: *SendStatistics*

Default value: *No*

Possible settings: *Yes, No*

Prepare statements on host

If this option is being used, function SQLPrepare will be sent to the host and a Precompiled Command will be stored on the host. This command can be executed as often as needed using SQLExecute. Using this option will result in performance gains when the same statement should be executed multiple times but with different parameter.

ODBC parameter

Key: *UsePreparedCommands*

Default value: *No*

Possible settings: *Yes, No*

Write trace file

If you select this option, the TCAODBC.LOG trace file is written into the tcACCESS installation WORK directory. This ouput is mainly needed for debugging purposes; it contains information about the operations that were performed by the tcACCESS ODBC driver.

ODBC parameter

Key: *Trace*

Default value: *No*

Possible settings: *Yes, No*

Parameter

The parameter described in this section are special purpose parameter.

The following parameter are supported:

/SQLID=creatorname

Using this option a standard value for the CREATOR (SQL-Id) in SQL-Engine queries can be specified. The CREATOR name specified will always be used when an SQL-Engine object is being referenced and no CREATOR has been specified in the SQL-statement.

The user id of the currently signed on user can be used as SQL-ID with keyword $$USERID.

Examples:

/SQLID=$$USERID

/SQLID=PRODENV

Processing: Parameter SQLID will be sent together with the first statement after the connection has been established. In the case that no SQLID has been specified or the SQLID has been nullified (/G: with no parameter) the statement will be appended by SQLID=;. Any valid SQLID will be changed at the frist execution of a statement after a successful Driver.connect().

During execution time the SQLID can be set using function SQLSetConnectAttr with SQL_ATTR_CURRENT_CATALOG. With the next to be executed statement, this new SQLID will be passed to the Host.

When saving a precompiled command with a specified SQLID this ID will be used during execution of the command independent of the currently active SQLID. It is not possible to reference precompiled commands unqualified that are stored in a TS queue even if an SQLID has been set.

The setting of the SQLID is only used to specify the creator if the object has been specified unqualified (Table, View, etc.) ans has no implication on security.

ODBC parameter

Key: *Parameter*

Return ODBC 2.0 version number

Specification of this switch will result in presenting ODBC Version "2.00" to calling programs.

Host-connection

You can specifiy the connection to be used for this ODBC DSN. The following can be specified:

Default:

The default slot or configuration weill be used as specified in the administrator settings.

Connection Slot:

The defined slot will be used

Configuration:

For the connection a slot will be selected, which is assigned to the defined configuration.

ODBC parameter

Key: *HostConnection*

Default value: *Default*

Possible settings: *Default, Slot, Config*

OK

Values entered in a dialog are verified and saved in the configuration file.

Cancel

Ends the dialog without performing any operation.

Help

Displays online help for the open dialog.

8.2 Definition of Applications with access to tcACCESS ODBC Driver

If the tcACCESS ODBC driver is used for certain applications only (e.g., in-house developed), it can be restricted by defining the TOAPAF.INI text file within the tcACCESS installation base directory. Each admitted application must have an entry in the [TOAPAF] section in the following format:

APPLICATION_NAME=LENGTH_OF_APPLICATION_NAME

Example:

```
  [TOAPAF]
MSACCESS.EXE=12
```

Note: 16-bit applications call the ODBC driver as KERNEL32.DLL within 32-bit operating systems.

8.3 Using the tcACCESS ODBC-driver as ODBC server

The Version 5 (and higher) of the tcACCESS ODBC driver can replace the tcACCESS ODBC Server Version 4. This mainly relates to supporting multiple connections to the host and "Connection Pooling". The ODBC driver is prepared to act as a data source for server applications (i.e. Web-Server).

The necessary steps are described in detail. •

Settings in the tcACCESS Main program:

- First, a configuration must be created in the tcACCESS main program, which contains the connection settings. Use the dialog "Connection Configurations" in the "Administrator" Menu.
- Next, use the "New" icon to create a pre-defined configuration file. Use a meaningful name for the configuration (i.e. "ODBC-SERVER").
- Next, use the dialog "Host connection" and dialog "Logon/Logoff" to define the corresponding entries. In the "Logon/Logoff" dialog supply the "**User Logon Defaults**" and activate option "**Show logon dialog**". This is necessary to avoid the display of a sign on dialog box for every connection to be established. Username and password will be encoded (as will all other settings under "Administrator") and will be saved to the configuration file.
- "Connection Pooling" is the utilization of existing but currently unused host connections. Using this pooling mechanisms reduces the overhead to establish and close a connection, which is normally necessary for each individual host request.

In order to use "Connection Pooling" activate option "**Close inactive connection after**" in dialog "**Various**". You can specify any time intervall, however it should not be less than 10 seconds. If this time interval expires an unused host connection will be closed.

- The last step is to invoke "**Connection Slots**" in the "Administrator"-Menu. Here you can assign the possible 255 connection slots to one or more configuration. Only connected slots can be used for data transmission.

  Based on your needs an the available tcACCESS Client licenses you can now assign multiple slots to the defined configuration. This determines the number of parallel host connections. If 10 parallel connections should be possible, you could, for example, mark slots 10-19 and assign these slots to the desired configuration using the "New" icon (i.e., assign to configuration "ODBC-SERVER").

ODBC-Settings:

- First you must create a tcACCESS ODBC data source using the Windows "Settings/Control Panel" (dialog ODBC Data Sources). By
- Clicking on the **Option>>** button the hidden settings will be displayed.
- If "Connection Pooling" should be used, activate option "**Allow deallocation of host session**". The tcACCESS communication handler is responsible for closing connections based upon the defined time inteval (refer to "**Close inactive connection after** ").
- Select a configuration in section "**Host Connection**" (i.e. "ODBC-SERVER").

When all settings are completed, this new data source may be used by server applications.

9. Supported Types of Host Connections

9.1 Third Party 3270 Host Emulation

The 3270 host emulation programs are used to establish the actual connection and communication. They provide program interfaces for control and use by tcACCESS.

The data may either be exchanged via the screen buffer on the 3270 emulation or by using the STRUCTURED FIELDS (SF) protocol. This connection type allows you to transfer data in binary form, directly within the 3270 data stream between host emulation and Host Communication Handler, and without having the need to use the screen buffer. Thus, higher transmission rates can be reached compared to the first method. Therefore, we recommend that the host emulation supports this type of protocol and that the appropriate PC Communication Handler is selected.

A description of the parameters are defined in Host Connections, (page192):

Session Id

Most of the 3270 host emulations allow you to establish several host sessions that are marked with capital letters (A, B, C, ...). This input field allows you to define the desired session. In most cases it is session "A".

Packet sizes

These two input fields display, if the selected connection type and/or the host emulation, supports the SF protocol which was described above. The input fields allow you to define packet sizes for data transfer between host emulation and the Host Communication Handler.

9.2 3270 Host Emulation by the "TCP/IP TELNET 3270" Communication Handler

This connection type uses the TELNET 3270 protocol to establish a session with the supported 3270 host environments (MVS, OS/390, VSE, VM).

The only requirement for the host is that a TCP/IP implementation (e.g., IBM, INTERLINK) with a standard TELNET 3270 server exists.

On the local PC, any type of WINSOCK implementation supporting at least version 1.1 may be used. With Windows 95 or NT and later, a TCP/IP support fullfilling this condition can be provided by using the operating system installation. Windows 3.1x offers this type of support with an available update for this purpose.

The PC Communication Handler contains a simple TELNET 3270 emulator that can communicate with the TELNET server on the mainframe. It is invisible to the

end user and is only used for logging on to the desired TP monitor and for data exchange, which is processed in the background (using the SF protocol, see Third Party 3270 Host Emulation above) between PC and host.

The advantages of the TCP/IP TELNET 3270 Communication Handler are:

- Fast and automatic session allocation
- No third party 3270 emulator required
- Easy installation on host and client platforms

A description of the parameters are defined in the section Host Connections, page 192.

Host address

This field must contain the TCP/IP address on the mainframe to be accessed. It can either consist of 4-dot separated number (192.168.0.1) or a name ("p390"). The input of a name is only possible when the TCP/IP configuration file hosts exists in the Windows directory (Windows 3.x, Windows 9x) or in its sub-directory "system32\drivers\etc\" (Windows NT) and holds an entry for it (e.g., 192.168.0.1 p390).

Port

This input field must contain the port number which the TCP/IP TELNET server on the host is connected to. By default it is the port '23'.

Packet sizes

These two input fields allow you to determine packet sizes for the data transfer between the TELNET 3270 emulator and the Host Communication Handler using STRUCTURED FIELDS (see also Third Party 3270 Host Emulation, page *1*).

Use SSL connection

If this option is active a secured connection to the host will be used (SSL - Secure Socket Layer).

LU name

If the mainframe to be connected has no terminal pool defined (e.g., for security reasons), a valid LU (logical unit) name must be specified for the identification of the logical terminal. This logical terminal is required for the TELNET 3270 emulator.

9.3 TCP/IP Connection

This Communication Handler establishes a direct session with the appropriate Host Communication Handler based on the TCP protocol.

IBM CICS SOCKETS are used for this purpose on the host (TCP/IP option for OS/390); for the local PC an optionally selected WINSOCK implementation is sufficient (TCP/IP support).

This communication type has the advantage that a session can be established quickly and very easy without requiring a logon script (see also example file tcalogon.sam) or a logon sequence (see Logon/Logoff, page *193*).

A description of the parameters are defined in the section Host Connections, page 192):

Host address

This field must contain the TCP/IP address of the mainframe to be accessed. It can either consist of 4-dot separated numbers (192.168.0.1) or a name ("p390"). The input of a name is only possible when the TCP/IP configuration file hosts exists in the Windows directory (Windows 3.x, Windows 9x) or in its sub-directory "system32\drivers\etc\" (Windows NT) and holds an entry for it (e.g., 192.168.0.1 p390).

Port

This input field must contain the port number to which the TCP/IP Host Communication Handler is connected. The default value for a tcACCESS installation is port "3020".

Use SSL connection

If this option is active a secured connection to the host will be used (SSL - Secure Socket Layer).

Start transaction

The name entered into this field describes the transaction on the tcACCESS Host Communication Handler. Normally, it should always be TCAS.

9.4 "Microsoft SNA Server" Connection

This connection type operates with three communication partners:

Host

The mainframe communicates with the PC server via the APPC protocol.

PC Server

The Microsoft SNA Server must be started on this server. It acts as a link between the host (via APPC) and the local PC (via an optional network protocol supported by Windows). The Microsoft SNA Server establishes a permanent APPC connection to the host which can be used from several SNA server clients at the same time.

Local PC

The client software of the Microsoft SNA Server installed on the local PC is connected with the PC server and is addressed by the tcACCESS PC Communication Handler.

A description of the parameters are defined in the Host Connections, page 192.

Local alias, Remote alias, Mode

These fields must contain the appropriate values of the MS SNA server configuration for the desired APPC connection.

Start transaction

The defined transaction name is sent to the APPC server on the mainframe where the transaction starts. Normally, this field should always contain TCAS (This is the transaction name of the tcACCESS Host Communication Handlers.).

The following example shows a valid configuration of the Microsoft SNA Server.

9.4.1 Example: Connection Settings ("DLC Properties")

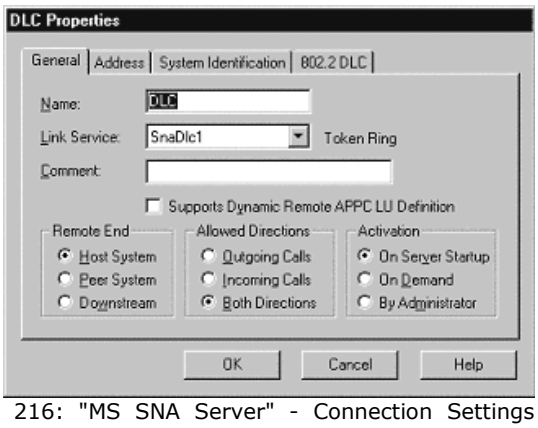


Fig. 216: "MS SNA Server" - Connection Settings (Part 1)

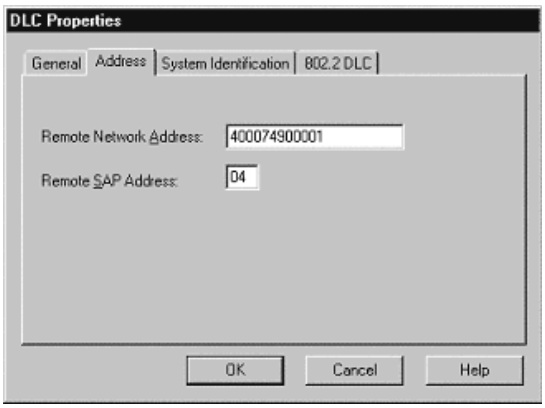


Fig. 217: "MS SNA Server" - Connection Settings (Part 2)

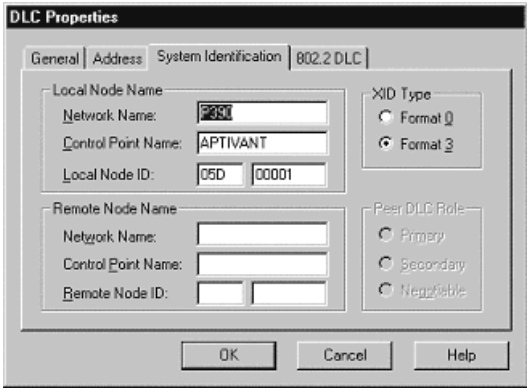


Fig. 218: "MS SNA Server" - Connection Settings (Part 3)

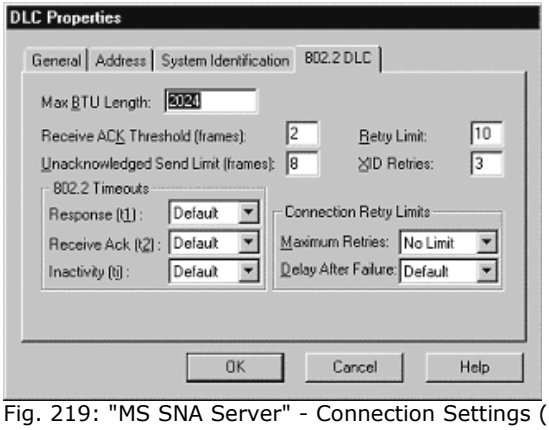


Fig. 219: "MS SNA Server" - Connection Settings (Part 4)

9.4.2 Example: Configuration of local APPC LUs ("SENDLU Properties")

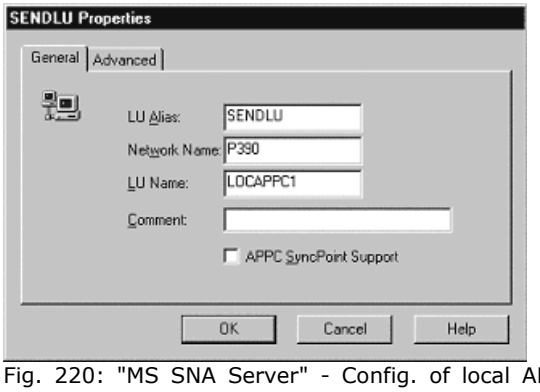


Fig. 220: "MS SNA Server" - Config. of local APPC LUs (Part 1)

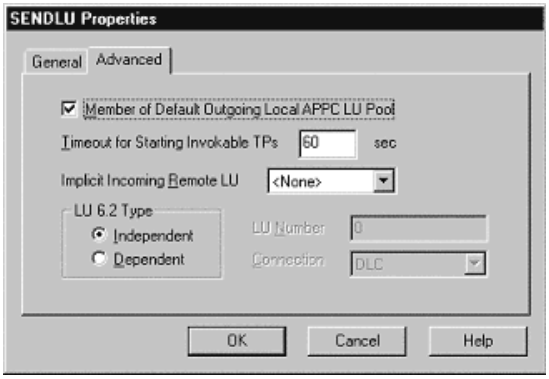


Fig. 221: "MS SNA Server" - Config. of local APPC LUs (Part 2)

9.4.3 Example: Configuration of remote APPC LUs ("RECVLU Properties")

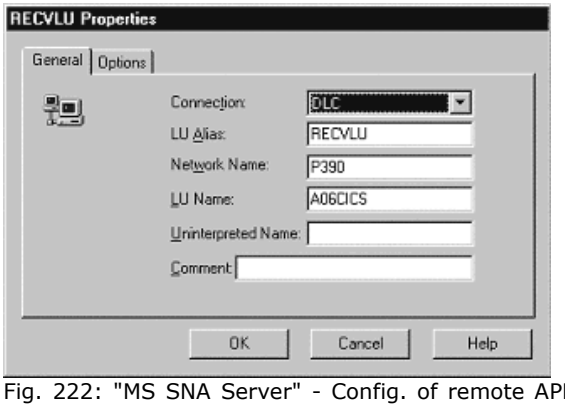


Fig. 222: "MS SNA Server" - Config. of remote APPC LUs (Part 1)

Fig. 223: "MS SNA Server" - Config. of remote APPC LUs (Part 2)

9.4.4 Example: APPC Mode Configuration ("#INTER Properties")

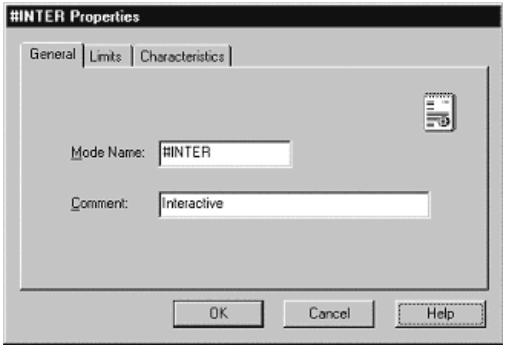


Fig. 224: "MS SNA Server" - APPC Mode Configuration (Part 1)

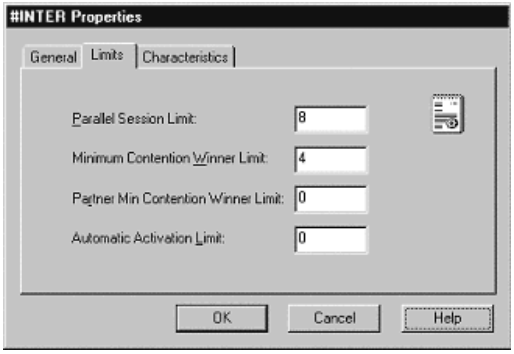


Fig. 225: "MS SNA Server" - APPC Mode Configuration (Part 2)

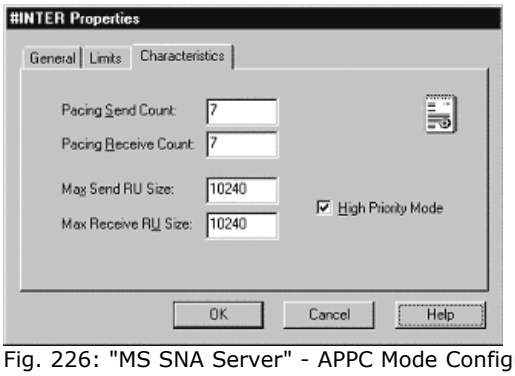


Fig. 226: "MS SNA Server" - APPC Mode Configuration (Part 3)

9.5 MQSeries Connection

If there is an MQSeries installation on the local PC and the selected host, you can use this interface for data communication.

A description of the parameters are defined in the section Host Connections, page 192.

Local queue

Name of the MQSeries queue that is to receive the responses from the host.

Remote queue

Name of the MQSeries queue that is to receive the requests to the host. It must either be

- A local definition of a REMOTE QUEUE (contains the name of the Remote Queue Manager), or
- A Local Queue of the Remote Queue Manager which is also to be defined.
-

Remote manager

Name of the Remote Queue Manager. If this value is not used, the defined Remote Queue must be a local definition. Otherwise, this name refers to the MQSeries transmission queue with the same name.

Max. message sizes

The message sizes used for the MQSeries connection are differentiated in incoming (Inbound) and outgoing (Outbound) packet sizes. They can be defined individually. The default value for these fields is 32000 bytes.

Start transaction

The name defined in this field is sent to the mainframe where the transaction starts. Usually this field should always contain TCAS (the transaction name of the tcACCESS Host Communication Handler).

Index