Kofax Communication Server

TC/REPORT Technical Manual

Version: 10.3.0

Date: 2019-12-13

KOFAX

Legal Notice

© 2019 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

Table of Contents

Chapter 1: Preface **6**
- TC/Report and TC/MA 6
- Reports 7
  - Report Generation 7
  - Report Formats 7
  - Report Delivery 7
- Billing 7

Chapter 2: Minimum Requirements **9**

Chapter 3: Structure of TC Report **11**
- Report Agent 12
- Fetch Agent 13
  - Fetching the Short Term Archive (Send Orders and Log Entries) 13
  - Fetching User Information 14
  - Fetching Application Downtime Information 16
- Database 16
  - Database and User Created by TC/Report Setup 16
  - Database and User Created by Customer 17
  - Physical characteristics of the database 25
  - Content of the Database 27

Chapter 4: Installation **46**
- Step 1 – Installation of the Database 47
  - Installation of MS SQL Server 47
  - Optional: Change Password of sa User 47
- Step 2 – Create KCS User Accounts 48
- Step 3 – Setup Agents and Database 52
  - Fetch Agent Setup 52
  - Report Agent Setup 59
- Step 4 – Configuration 62
- Cost Accounting 62
- License for TC/Report 63
- Final Installation Steps 63
- Configuration for KCS Tandem System 63
- Configuration for Country Codes (rr99) 63
- Additional Configuration for Application Monitoring 64

Chapter 5: Configuration via TC Management Console..... 67
Windows Server 2008 or Later..... 68
Fetch Agent Configuration..... 68
Servers Property Page..... 68
General Property Page..... 76
Database Property Page..... 77
SLA Monitoring Property Pages..... 79
Report Agent Configuration..... 93
Chapter 6: Special Features of the Fetch Agent..... 95
Handling of Media Servers in ASP Systems..... 95
Handling of LCR (Least Cost Routing) Systems..... 95
Handling of Incoming Fax or Telex..... 97
Automatic Deletion of Oldest Records..... 98
Backup of the Database..... 99
Calling a Batch File When Backup Is Ready..... 100
Sending a Message When Backup Is Ready..... 101
Requirements and Restrictions of the Backup..... 101
Basic Integration with UTC-Based TCOSS..... 102
Chapter 7: Maintenance..... 104
Performance Counters..... 104
Scripts for Manual Database Backup and Cleanup..... 106
TIMED_BACKUP.BAT..... 106
TIMED_CLEANUP.BAT..... 107
Combining TIMED_BACKUP and TIMED_CLEANUP..... 107
Database Upgrade to Unicode..... 108
Upgrade TC/Report Database..... 112
Fetch Agent Registry..... 114
FetchWorkStations Registry..... 118
Report Agent Registry..... 119
SSL/TLS Registry Keys..... 122
Accessing the Database..... 123
Error Handling..... 126
Chapter 8: Creating or Modifying Reports..... 129
Setting Connection Parameters for New Reports..... 129
Report Options..... 132
Setting Table Locations When Modifying Reports..... 133
Parameter Fields..... 134
Special Parameters for Integration with TC Management Console..... 134

INI Files for Standard Reports..... 136
Translating Existing Reports..... 138
Recommendations for CSV Reports..... 138
Chapter 9: Hints..... 140
Do Not Change the SQL Query for the Reports..... 140
How to Count the Number of SMS Sent..... 140
Event Log Warning with SQL 2005..... 140
Using TC/Report with Non-Standard Languages and Character Sets..... 141
Configurable String Separator for CSV Report Format..... 142
Failure Audit Log Entry During Report Creation..... 143
Calculated Recordsizes in Fetch Agent Trace..... 143
Chapter 10: Performance..... 144
Chapter 11: Results of Performance Tests on a TCOSS Server..... 146
Used Test Environment and Modus..... 146
Environment..... 146
Modus..... 146
Test Results..... 147
Configuration and Results..... 147
Results Sorted by Database Options..... 149
Recommended Values for the Tested Database Settings..... 152
Performance Loss Under Recommended Database Configuration..... 152
Raw Test Results..... 154
Chapter 12: Restrictions and Significant Limitations..... 155
Chapter 13: Usage Example: An ASP Billing Interface..... 158
Installation..... 158
Install Additional Files..... 158
Create Output Directories..... 158
Generate and Configure TCReportXML Table..... 159
Modify the Registry for TCReportXML..... 161
Modify the Registry for TCSRV Support..... 162
Registry..... 163
Manual File Creation..... 165

Chapter 1

Preface

TC/Report is a Kofax software product that was designed to keep track of the message traffic within a company. It creates a report for a desired time span that lists all messages sent, who sent them, the service used, the size and the cost incurred.

TC/Report provides flexible statistics and accounting for Kofax Communication Server. It can generate reports in many formats and deliver them to any type of recipient.

TC/Report also provides a very simple and open interface to 3rd party billing systems. Data can be preprocessed and sent to the billing system in many formats.

In addition, data from 3rd party systems can be imported and integrated into one unified report. Currently, importing of 3rd party data requires customization by Kofax staff.

The system collects data from configured KCS systems, stores and processes it, and creates output according to the customer's needs.

There are three basic uses of the system:

- **Statistics** – used for general statistics. Create reports to monitor the overall performance of the Kofax Communication Server (e.g. line usage, message traffic, message distribution). The product includes several highly configurable reports for all kinds of statistic.
- **Accounting** – used for calculation of costs (for users, groups, cost-centers).
- **Billing** – used to export the accounting data to external billing systems.

Reports usually contain graphs and tables in user-friendly form. They can be sent to any mail recipient or printed on a defined network printer.

Important ! The Kofax Communication Server and its components formerly used the name TOPCALL. Some screen shots and texts in this manual may still use the former name.

TC/Report and TC/MA

This document is about reports created by the TC Report Agent via Crystal Reports.

Another set of reports will be available via the product **TC Messaging Analyzer** (TC/MA).
TC/MA uses the TC/Report Fetch Agent for data collection. Reporting is done via the product OmniAnalyser from HyperSoft.

Several database fields and configuration settings implemented for TC/MA integration are mentioned in this document. A separate TC/MA manual describes the product in detail.

Reports

The system comes with a set of standard report templates. Each of these templates allows various filters (users, user groups, message classes, etc.). This way, various reports can be generated without the need to modify the standard templates.

Report Generation

Reports can be automatically created and scheduled (daily, weekly, etc.).

Report Formats

Report messages can contain attachments with the following formats:

- CSV – Comma-separated values (this file format does not support graphics)
- RPT – Crystal Reports
- XLS – Excel
- HTML 4.0
- RTF – Rich Text Format (to be viewed with Word for Windows)
- DOC – Word for Windows
- PDF – Portable Document Format

Report Delivery

Reports can automatically be printed to any network printer, sent to any recipient (can also be a 3rd party application like a billing system) or posted to the web as html.

One unified view for all types of activity:

The system treats the traffic on all services (links, TC/Voicemail, fax, telex, X.400, SMTP, SMS) in the same way.

One unified view for many servers:

One view for multiple TCOSS Servers.

Any statistics possible: costs, calls, errors, etc. **per**: user, user group, cost center, etc.

Billing

This feature allows to automatically export data into customers billing system. Some reports are especially designed for billing purposes. The TCOSS administrator can select them during setup to improve customers billing system functionality.

As a general, rule 3rd party billing systems can import data formats supported by Crystal Reports:

- CSV – Comma-separated values

- XLS – Excel

To interface to a billing system:

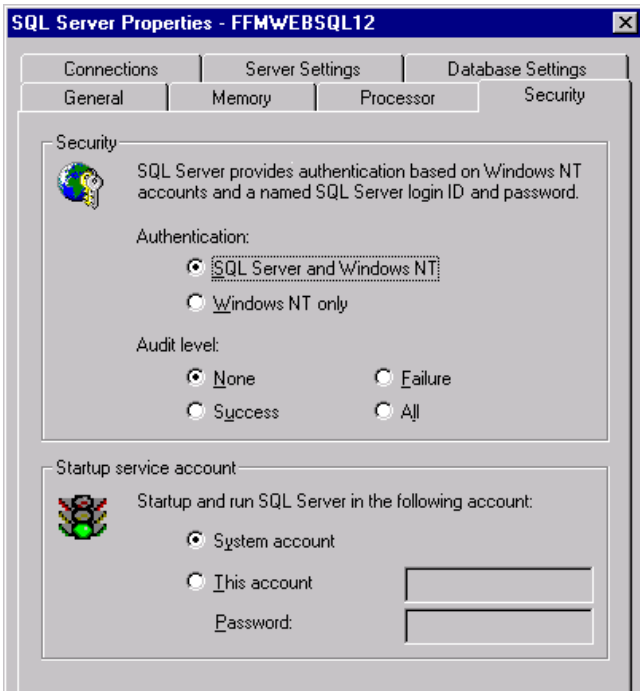
- Define a report with output as data file to a billing system, in the format that the billing system requires
- Set up an automatic schedule to send the report to the billing system

Chapter 2

Minimum Requirements

TC/Report requires:

- KCS Monitor (delivered with KCS)
- MS SQL Server Express Edition (license free) or MS SQL Server (license supplied by customer).
  Supported versions: 2005, 2008, 2008 R2, 2012, 2014, 2016 and 2019.
  The SQL server must have the SQL server authentication (and not the Windows authentication).



- Microsoft ActiveX Data Objects (ADO) 2.5 or later. ADO 6.0 is a part of Windows Server 2008 R1. ADO 6.1 is part of Windows Server 2008 R2.
  TCReport uses the SQLOLEDB provider that is part of ADO.
- The required runtime modules of Crystal Reports are installed with KCS setup.
  No Crystal Reports license needed, except for running the Report Agent as a provided service for other companies.
  Printer – if used for reports

- For fax output: KCS Fax Printer driver to convert a Report to the KCS Image format (TCI). The printer driver is installed with TCfW. If TC/Report shall run on the TCOSS server, install TCOSS before installing TCfW.

  Note For operating system requirements, please refer to the Platform System Manual.

TC/Report works with:

- TCOSS Release 7.27.00 or higher
- for some features, higher TCOSS versions are needed (e.g. TS_Channel: TCOSS Release 7.38.00 or higher and TCSI 2.33.00 or higher, send retries + log entries: TCOSS 7.41.00). See the Action_Table description (see Content of the Database) for detailed information.
- TCfW version 5.01.01 or higher is needed for Distributor statistic, to configure queue length logging and to set the new Report right.
- TC/Broadcast from KCS 7.46.13 or higher is needed for the TCDL and TCJOB reports. All broadcast user profiles must be configured correctly, especially the job start and job end events. TCNOTIF must be running.

TC/Report requires a license on each monitored TCOSS instance. The license can be entered via licences.exe (part of KCS).

TC/Report can be installed on the computer where TCOSS is installed. For best performance, we recommend to run no other processes (e.g. Links) on the same machine.

Hard disk space requirements of TC/Report are 400 MB plus two times the configured maximum size of the database (see Step 4 – Configuration).

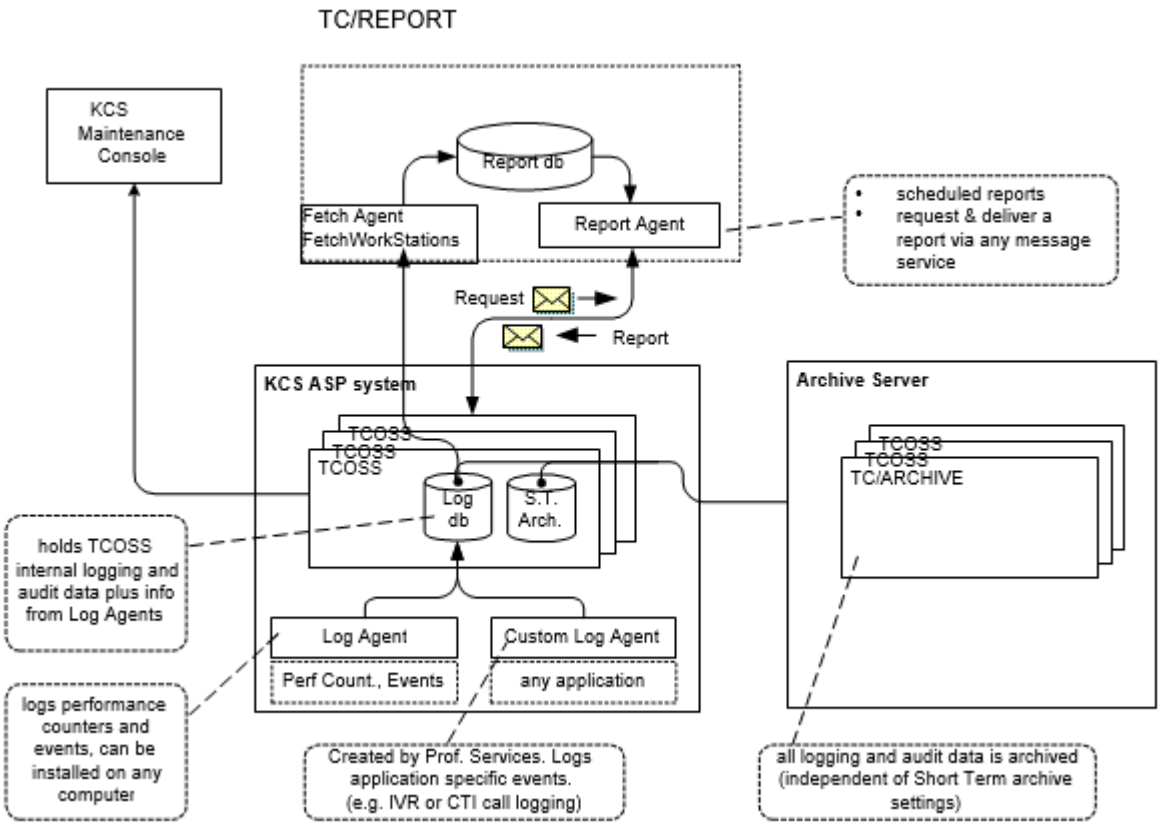
Cost accounting must be configured on TCOSS for fax and telex; costs must be configured in links that shall be used for accounting. *(Please refer to TCOSS manual and TC/LINK-manual for detailed information.)*

Entries must be archived in the short term archive in order to be seen by the fetch agent. Send orders that are not available in the short term archive are not part of the statistical information. The long term archive has additional filter possibilities, so there may be more send orders archive (short term) for TC/Report than are written to the (long term) Archive server.

The license free MS SQL Server Express edition has a 4 GB size limit for databases. This is enough for approximately 3.600.000 short term archive entries. Customers who need to avoid size limitations can use MS SQL Server without changes in TC/Report product. For upgrading, use the Copy Database Wizard from MS SQL Management Studio.

Chapter 3

Structure of TC Report

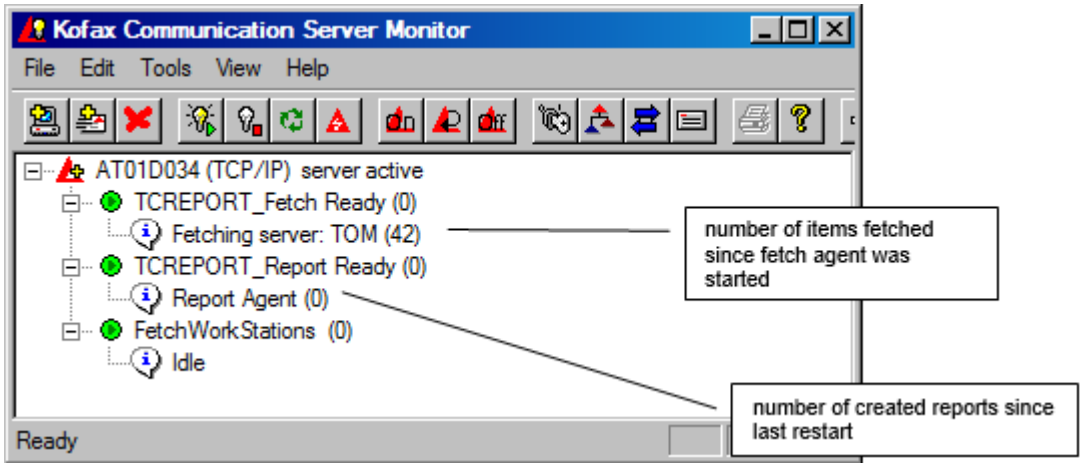


TC/Report is an application controlled by TC/SRV. It consists of 2 main agents: The Fetch Agent and the **Report Agent**. Only 1 **Fetch Agent** process is allowed per database. If application downtime fetching is configured, this is done by a third process (**FetchWorkStations**).

This is how the agents show up in the KCS Monitor:

number of created reports since last restart

number of items fetched since fetch agent was started



Report Agent

The **Report Agent** polls its KCS inbox to fetch a command message, a message containing the instructions on what information the report should contain. Once it found a message of this sort, it starts Crystal Reports to create the report specified in the command message. This report is then automatically sent to the person (may also be a fax, a printer, or a file) specified in the command message (by default the person requesting the report). After successful sending the status of the message in the queue is set to **completed**. If the report runs into an error, the command message is acknowledged negatively and the TCOSS retry procedure will handle retries.

The tasks of the Report Agent are:

- Poll an inbox on the TCOSS server for command messages
- Read all parameters for the report from the message
- Check if the person requesting it is allowed to create reports

> **Note** This person needs either the **Reporting** right or the **All Users – List inbox** and **List outbox** right

- Start Crystal Reports to create the report
- Create a report based on the parameters in the command message Report Agent will:
  - Create a standard KCS message with the report as an attachment (the message can be sent to the customers billing system) or
  - Print the report on the defined printer (if specified in the command message) or
  - Store the report in the file system (e.g. as HTML on a web server)
- In case of periodical messages: Create a follow-up message for the next scheduled time
- Store INI files for the report request agent in the FIS folder

Fetch Agent

To be able to create a report the Report Agent needs data. Most of this data is collected by the Fetch Agent.

Database content

The greatest bulk of data stored in the database is information about TCOSS send orders. The Fetch agent collects this data from the short-term archive of one or more TCOSS instances. Depending on configuration, the Fetch agent collects terminated messages only (that is the default), or also send attempts (there may be several send retries until the message is terminated).

For further classification of the fetched send orders, TC/Report maintains a channel table (mapping addresses to services) and a server table (mapping TCOSS instances to customers).

Additionally, the TC/Report database can contain log entries. These are application-specific data that are also read from the short term archive. E.g., information about the status of KCS queues, TC/Broadcast jobs, etc.

You can also configure the Fetch Agent to store basic information about KCS user profiles. This makes it possible to assign send orders to users, user groups, departments, and companies.

Optionally, the TC/Report database can contain information about application availability. For this purpose, additional tables hold data about workstations and the applications they are hosting.

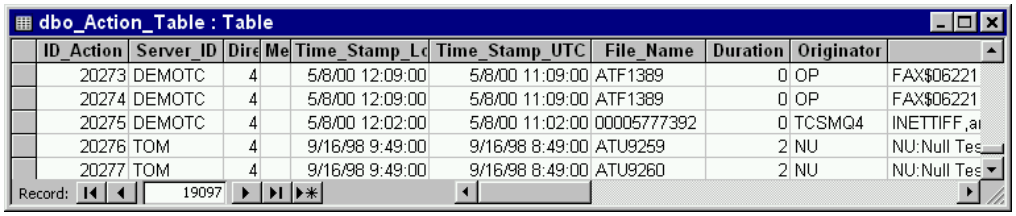
If the *TC/Probe* is installed, there is another set of tables with information about periodic test messages sent by this agent.

Fetching the Short Term Archive (Send Orders and Log Entries)

The Fetch Agent reads the send orders from the TCOSS short term archive. Short interruptions of the Fetch Agent's connection to TCOSS (e.g., by stopping Fetch Agent or TCOSS) do not matter, because the Fetch Agent afterwards resumes fetching from where it left off.

During Setup, you define a poll interval and the standard parameters used for the connection to the Kofax Communication Server.

Messages of several TCOSS instances can be stored in the same database.
Of course, it is also possible to have several Fetch Agents, each of them polling only 1 TCOSS instance and writing to its own database (e.g. for ASPs).



The Fetch Agent stores the send orders in the **Action_Table**. Date and unique ID of the last fetched entry are stored in the **Server_Table**.

Terminated Messages

With standard settings, the Fetch agent retrieves only terminated messages from the short term archive. As an alternative (see next paragraph), all send attempts can be fetched.

Send Retries

TCOSS version 7.41.00 and above stores all send attempts (successful or not) in the short term archive. TC/Report can be configured to use this new feature. This makes new report types possible (e.g. line statistics including send retries).

Of course, storing all send attempts in the database increases the database size and also has an impact on TC/Report performance. Therefore, it must be explicitly enabled: In the TC Management Console, you can define per server if all send attempts shall be stored in the database. If the TCOSS version of the server is below 7.41.00, the default setting (only final send attempts stored) is used.

Reports delivered with previous releases (UNIVERSAL, UNIVERSAL_GRAPH, RPT*) may only be used if the new feature is disabled. Otherwise, they yield wrong results (too many messages).

The new reports are aware of the new feature. Statistic results based on send attempts (e.g. number of send retries) are only available if the Fetch agent is configured to store send attempts.

If there are several retries for a send order, cost of retries are accounted to the period they fall into.

Log Entries

The TCOSS Log Agent (optional feature of TCOSS version 7.41.00 and above) creates periodic logs of a server's health state. An example is the TCOSS queue length agent.

TC/Report can be configured to include these logs in its SQL database. Logging information is stored in a separate table **Log_Table**.

This allows creation of reports showing combined graphs of server health conditions for a distributed system. The first report of this kind was the Queue Length report. In the meantime, there are other applications that create log entries, e.g. TC/Broadcast. As all log entry types are stored in the **Log_Table**, additional log agents and corresponding report types can be created by third party developers without changing TC/Report.

In the TC Management Console, you can define per server if log entries shall be stored in the database. If the TCOSS version of the server does not support logging, TC/Report uses the default setting and does not try to fetch log entries.

The TC/Report database has a limited size, which can be configured in the TC/Management Console. If the maximum size of the database is reached, entries will be deleted from the Action_Table and from the Log_Table.

Fetching User Information

For every TCOSS instance, you can define whether user profile data shall be exported to the TCREPORT database. No user information is fetched from disabled servers.

Currently, only the TCLINESN report evaluates the user information.

When the Fetch Agent logs in to KCS for the first time, it reads all user entries and updates the user table accordingly. Reading all user entries may take some time: up to 5 minutes for 1000 users.

Subsequently, user profile updates are fetched at every poll cycle. The Fetch Agent stores the starting point for the next user profile fetch (i.e., 2 values, for user store and recipient store) in the database.

Under special conditions, the TCOSS clears its internal change buffers and the stored starting points are not valid anymore. Such conditions are: restart of TCOSS and overflow of the internal change buffers. If this happens, the Fetch Agent will have to read all existing user entries once more.

It is possible to configure an additional daily full fetch.

The Fetch Agent checks the configuration for user profile updates at every poll cycle. If you disable user profile data fetching, the existing information will not be deleted, it will just not be updated anymore.

Full fetch:

All KCS user profiles are read. This is done after first login to a server, after TCOSS restart and (optionally) once a day at a configured time.

New users: new entries are added to the User_Table, DateValid is set to the current TCOSS date and time, DateInvalid is left empty.

Changed users: For users that already exist in the table, the Fetch agent checks if the information stored in the current record has been changed. Only if information has changed, the old record is invalidated by setting the DateInvalid to the current TCOSS time and date, and a new record with the current information is added to the table.

Deleted users: A full fetch yields no information about deleted users. Therefore, records referring to user profiles that are already deleted on KCS remain intact.

Update fetch:

The Fetch Agent gets only information about user profile changes. This is done at every poll cycle of the Fetch Agent.

New users: new entries are added to the User_Table, DateValid is set to the current TCOSS date and time, DateInvalid is left empty.

Changed users: For users that already exist in the table, the Fetch agent checks if the information stored in the current record has been changed. Only if information has changed, the old record is invalidated by setting the DateInvalid to the current TCOSS time and date, and a new record with the current information is added to the table.

Deleted users: Records for deleted users are invalidated by setting DateInvalid to the current TCOSS time and date.

Refetch:

In the TC Management Console, the **Refetch** checkbox in the Fetch Agent Server panel deletes all table entries concerning this server. Subsequently, the Fetch Agent does a complete refetch of user information.

Delete:

Deleting a server in the TC Management Console removes all user entries from this server from the **User_Table**. No more user information is fetched for this server.

Fetching Application Downtime Information

Optionally, TC/Report reads new event log entries of configured workstations once a day. This is done by a dedicated process: **FetchWorkstations.exe**. For workstations that have been added newly, only the event log entries from the last few days are read (the exact number of days can be configured, default is 3). This is done for performance reasons, - reading the complete event log can take a very long time.

The software evaluates the retrieved event log entries to find out when an application was started or stopped.

Some applications do not have a dedicated “application start” event. For these applications, you have to configure TCSRV to create an “application start” event. You can do this by setting registry key *HKLM \Software\Topcall\Boot\ProcessReady* to 1 on all involved servers. TCSRV will then log the start of any KCS application on this workstation. In the table of application types (section *SLA Monitoring Property Pages* of this manual), the third column shows if the application needs the TCSRV ProcessReady event.

The number of days fetched for new workstations is configured via TC Management Console, in the General panel of the Fetch Agent.

Application status changes are written to the table **App_Downtime_Table**. Please note that this table holds all information that is needed for the Application Downtime Report: the application name, the application group it belongs to, and the customer name. The report reflects the application hierarchy of the time when TC/Report detected that the application was unavailable. If you assign the application to another group or customer later, the new information will only affect application status changes that are detected afterwards.

Database

TC/Report uses its own database, and logs on to this database with a dedicated SQL server user who is the owner of this database. Database and user are normally created during setup of the TC/Report Fetch Agent. For this purpose, setup needs the credentials of an SQL Server login with permissions on the master database. For installations where master database access is not allowed (because the SQL server is used for other purposes also), this chapter describes how the TCReport database and its owner can be created by the customer, before installing the Fetch Agent.

Database and User Created by TC/Report Setup

In a standard installation, the TC/Report database and its owner are created automatically during Fetch Agent setup. For this purpose, Setup needs the SQL server credentials of a user who has full access to the master database and is allowed to create databases. This may be the built-in “sa” user or another user with similar rights.

Setup grants the TC/Report user access to the master database and gives him the right to create databases. This is needed for the creation of backup databases.

Database and User Created by Customer

The customer can create the TC/Report database and the TC/Report SQL user before TC/Report installation. This section shows how this is done via MS SQL Server Management Studio (screen shots are from MS SQL 2008 Express with Tools).

Advantage: The customer can control the size of the database (e.g. provide a fixed-size database) and the access permissions of the TC/Report user (e.g. deny him access to the master database).

Disadvantages:

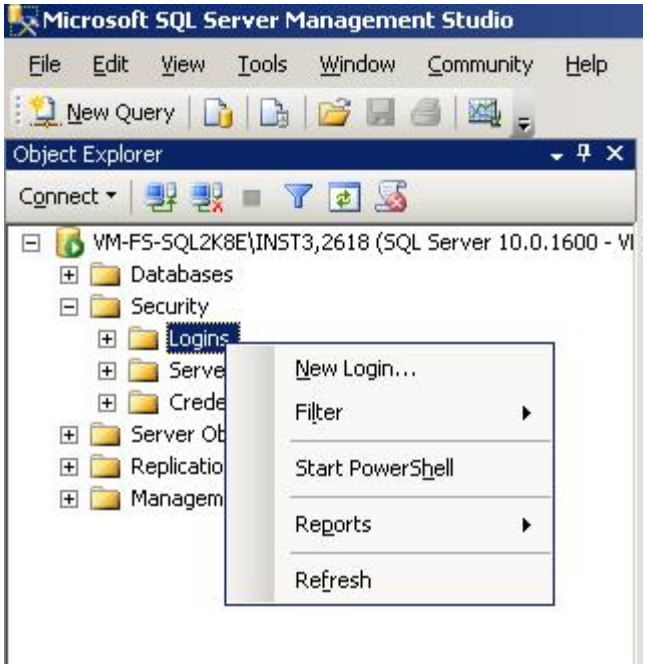
If automatic backup databases are needed, they must also be created manually before activating backup via TC Management Console.

Due to missing permissions, it is not possible to configure the maximum memory usage of the SQL server via TC Management Console.

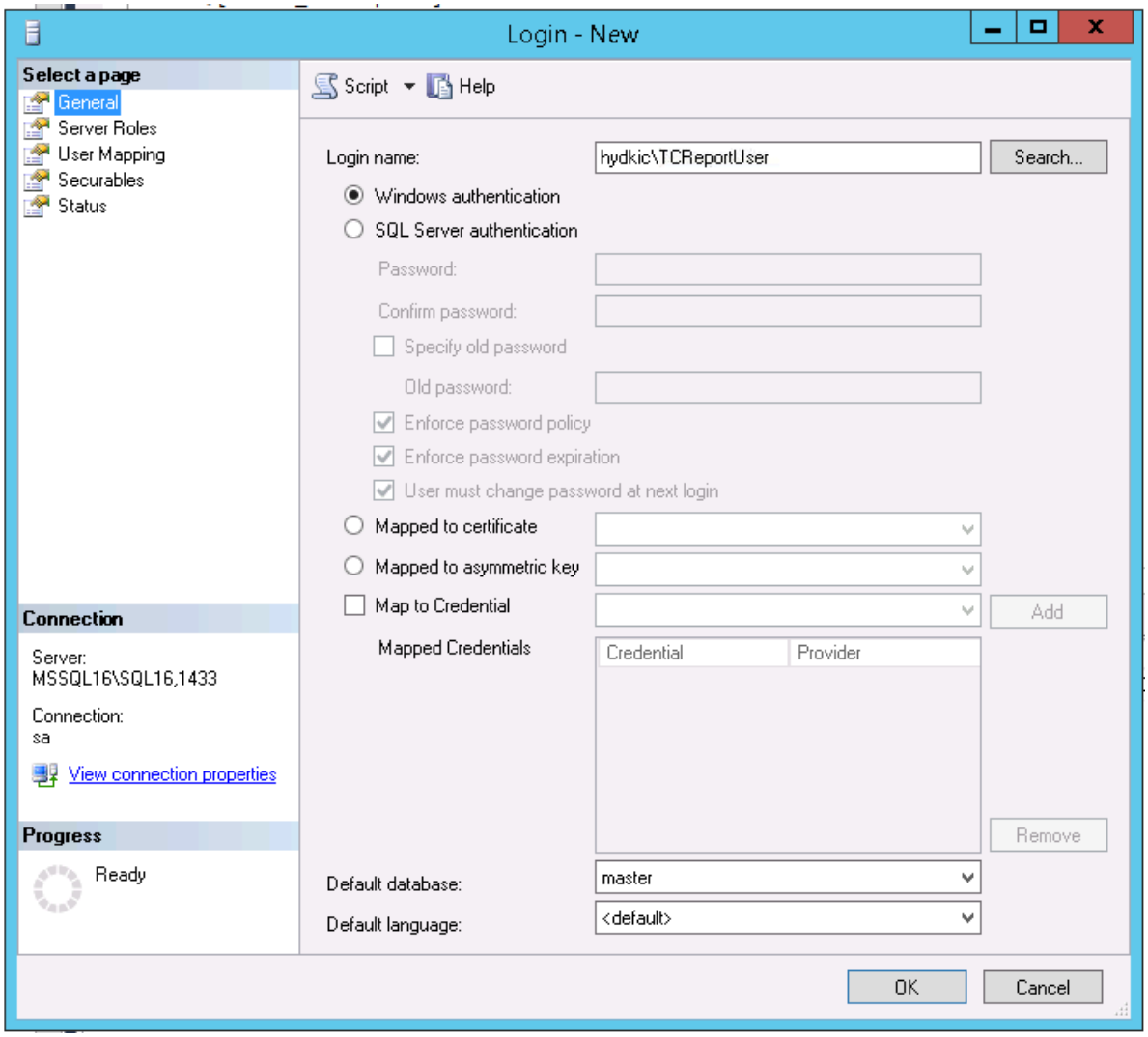
Create the TCReport User

This section explains how to create the SQL Server login used by TC/Report via the MS SQL Server Management console. In this example, the login is called TCReportUser.

After connecting MS SQL Server Management Studio to your SQL server (using a sysadmin account), right click the Logins Node below the Security container, and choose “New Login” from the context menu.



The **Login – New** property page appears. You need to edit the **General** section here.



In **Login name**, enter the name of currently logged in Windows user (domain user) and select the **Windows authentication** option,

Note Add the Windows user to properties of the Login as a service policy. This policy is available in the **Local Security Policy** screen.

As TCReport Fetch Agent will run as a service, it is recommended to disable the options “Enable Password Expiration” and “User must change password at next login”.

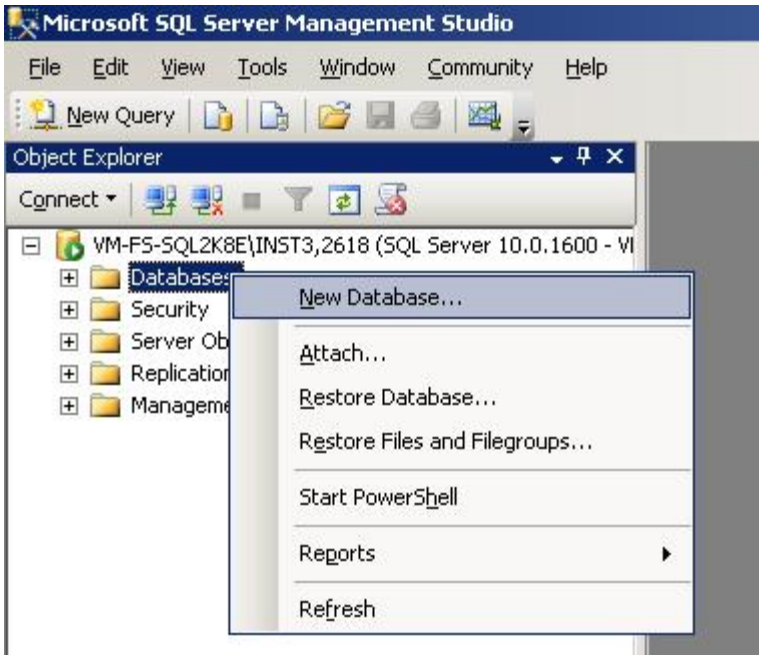
Leave the remaining settings at their default values, and click **OK** to save the new login.

Note After creating the TCReport database (next step), you can change the Default database field to the name of the TCReport database. The effect of this change is that when you log on via osql or sqlcmd as this user, you are already logged in to the TCReport database.

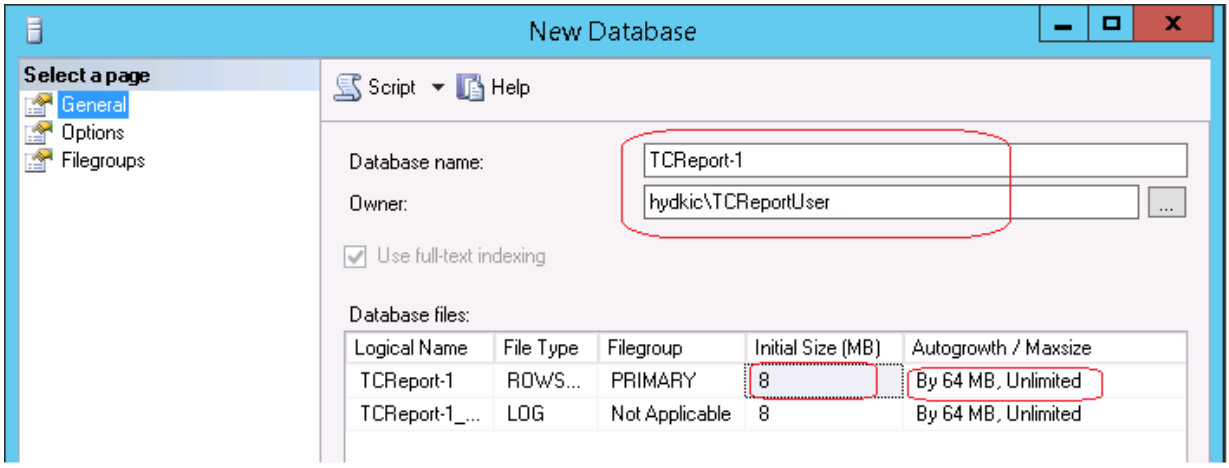
Create the TCReport Database

Now that the TCReportUser exists, you can create the TCReport-1 database.

In SQL Server Management Studio, right-click the **Databases** container and choose context menu option “New Database…”.

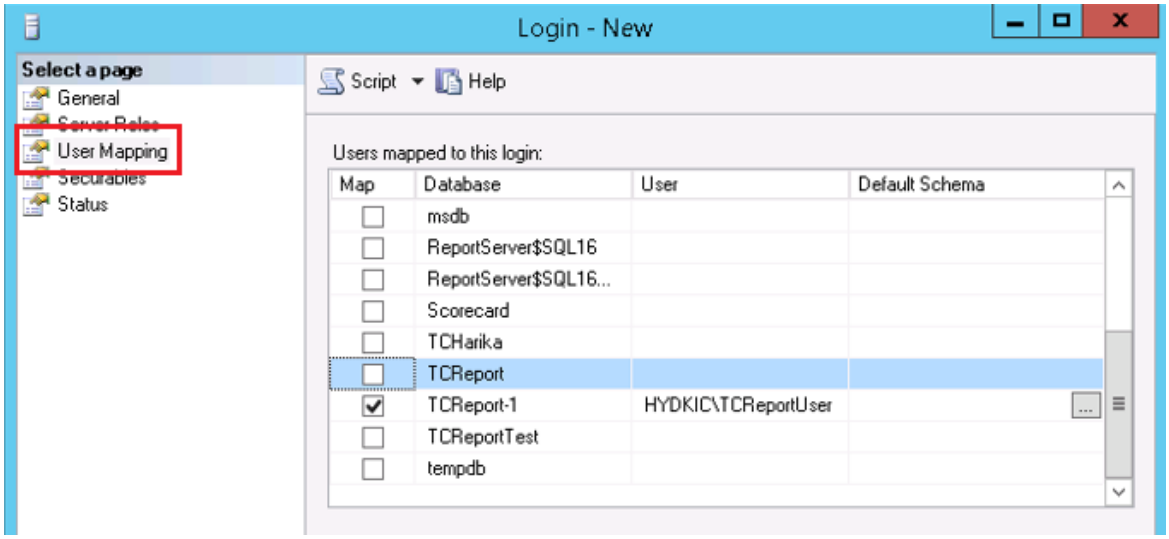


The property page of the new database appears.

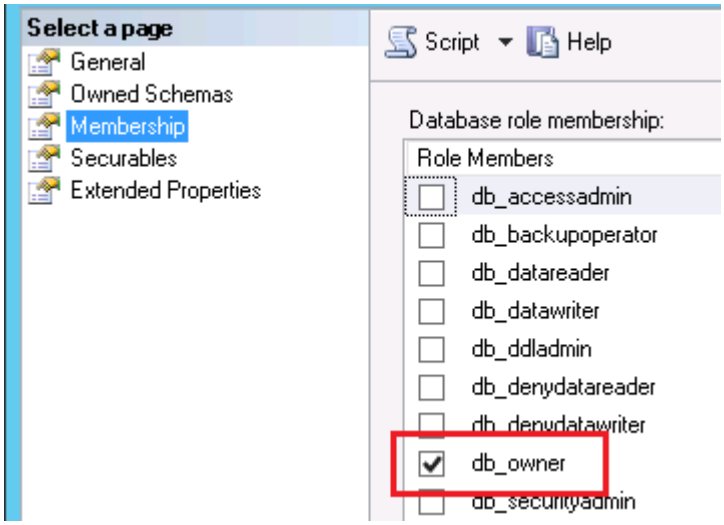


In the **General** section, enter the name of the new database (TCReport-1) and enter the login (TCReportUser) as the database owner.

In the **User Mapping**, make sure that the database is mapped to the selected user.



Navigate to the newly created database and select the **Security** option. Make sure that the **db_owner** role is assigned to the database user.



By default, the database consists of two files, the actual database file (TCReport.MDF) and a transaction log file (TCReport_log.LDF). Changes to the database are recorded in the log file.

You can change the initial size of each file, and the options for automatic growth. The above example shows the settings that are typically used for TC/Report.

As an alternative, the customer can create the database with a fixed size and disable automatic growth:

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth
TCReport	Rows ...	PRIMARY	1,900	None
TCReport_log	Log	Not Applicable	10	By 10 percent, unrestricted grow...

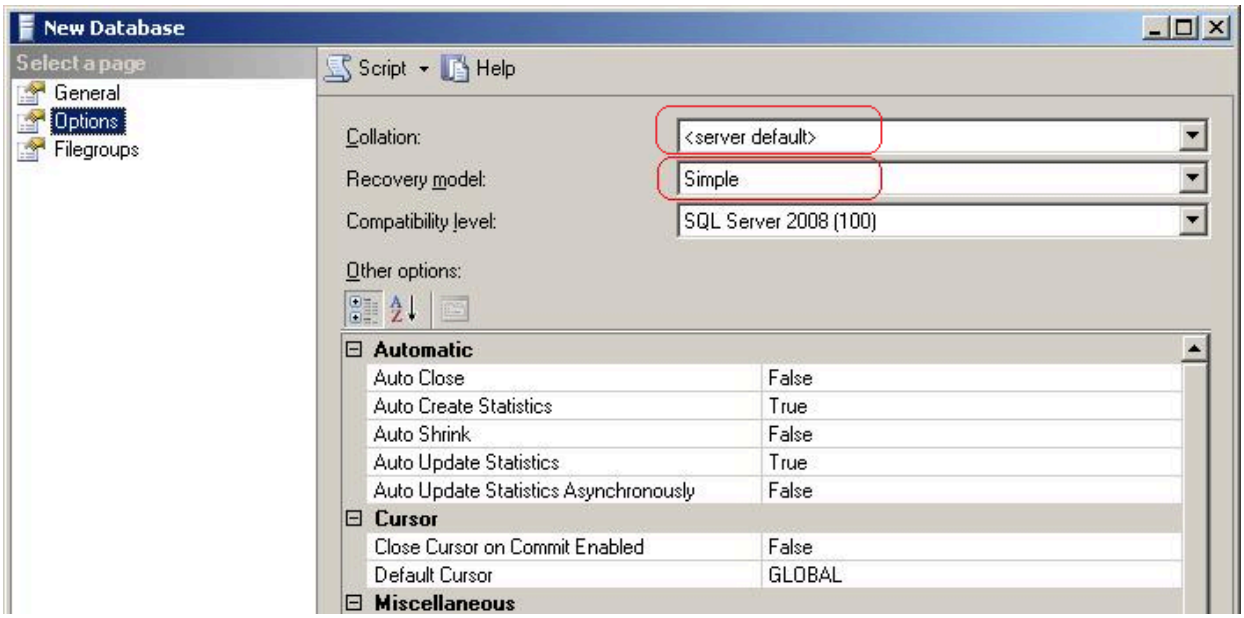
Attention:

With a fixed size database, automatic deletion of oldest entries works only if you use the option “Content Size” in the TC Management Console database panel (or directly write to registry value DBMaxDataSize). See also section Database Property Page.

The General settings of the database contain also the path of the database, i.e. the folder where the database files are stored. The path can be modified.

Logical Name		Path
TCReport	...	c:\Program Files\Microsoft SQL Server\MSSQL10.INST3\MSSQL\DATA
TCReport_log	...	c:\Program Files\Microsoft SQL Server\MSSQL10.INST3\MSSQL\DATA

In the **Options** section of the database properties, set Recovery model and Collation according to your needs. All other settings must remain at their default values!



Optional: Create Backup Databases

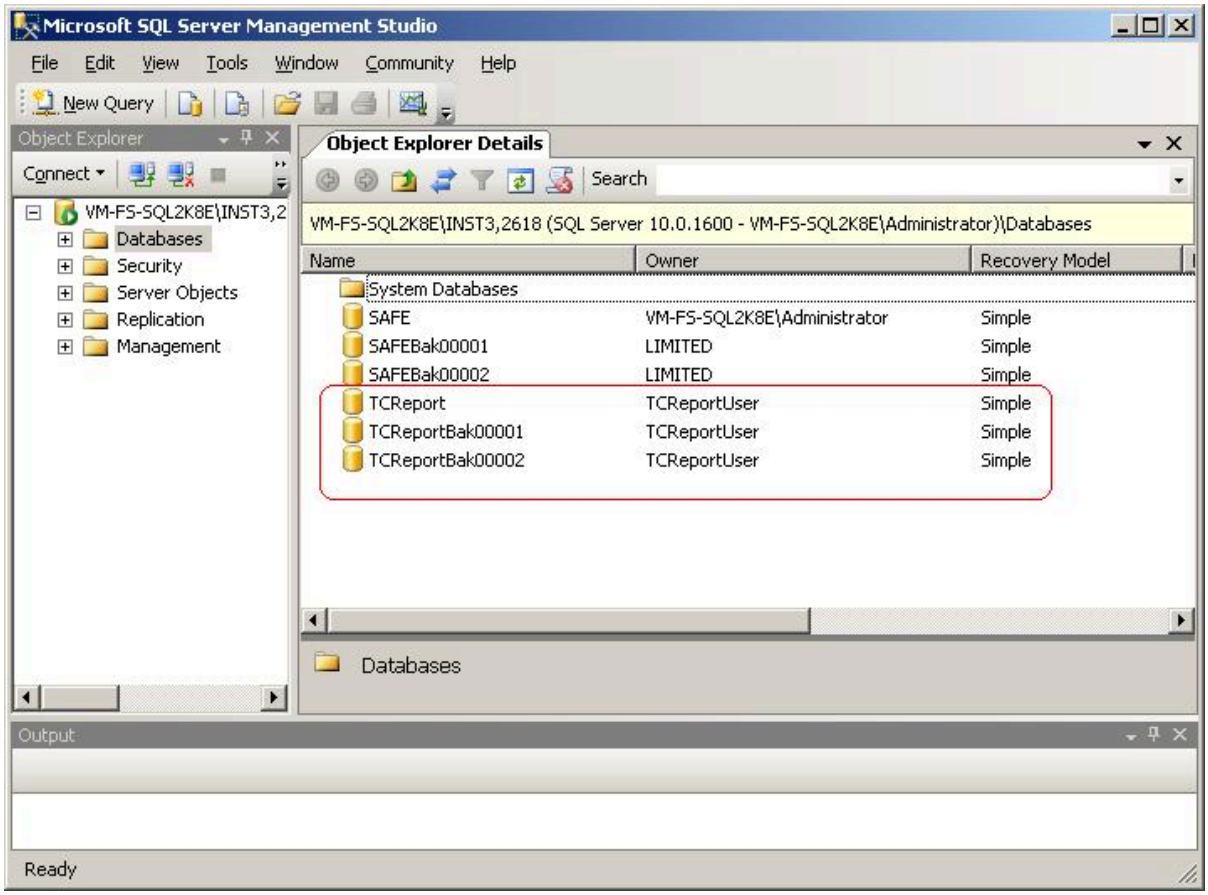
If TC/Report shall write to backup databases, all backup databases must be created manually before activating backup via TC Management Console. This is done using the same method as described in the previous section.

Backup databases should be unrestricted in growth. Their logical names must consist of the logical name of the main TCReport database (e.g. “TCReport”), followed by “Bak” and the 5 digit backup number (from 00001 to the configured number of backup files).

Database owner, path, collation, recovery model must be the same as configured for the main TCReport database.

The next screen shot shows the details for an installation with 2 backup files. (To see the Object Explorer Details screen, select menu item View | Object Explorer Details in the menu line of Microsoft SQL Server Management Studio.)

Example:

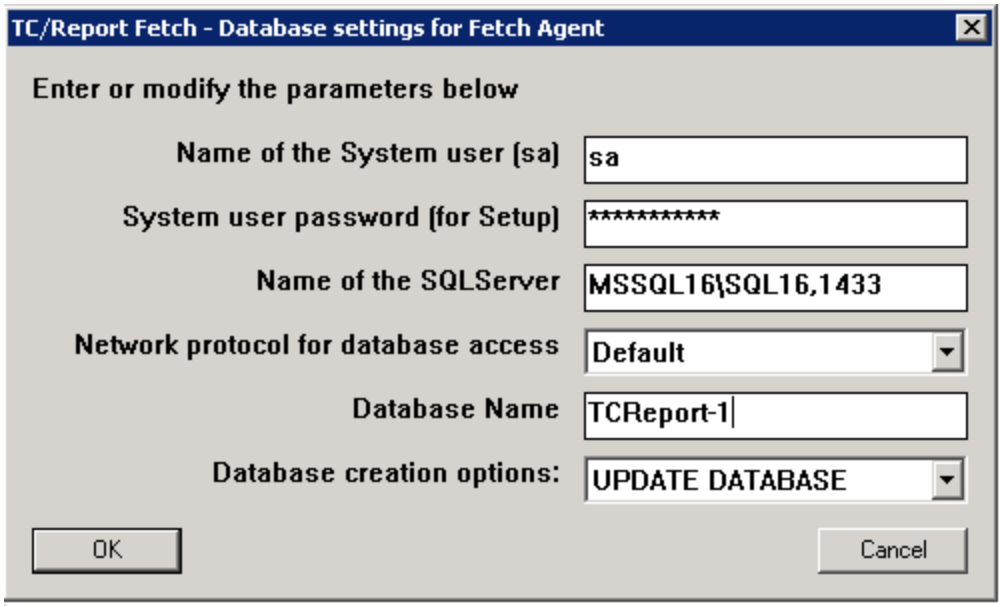


Hints for Fetch Agent Setup

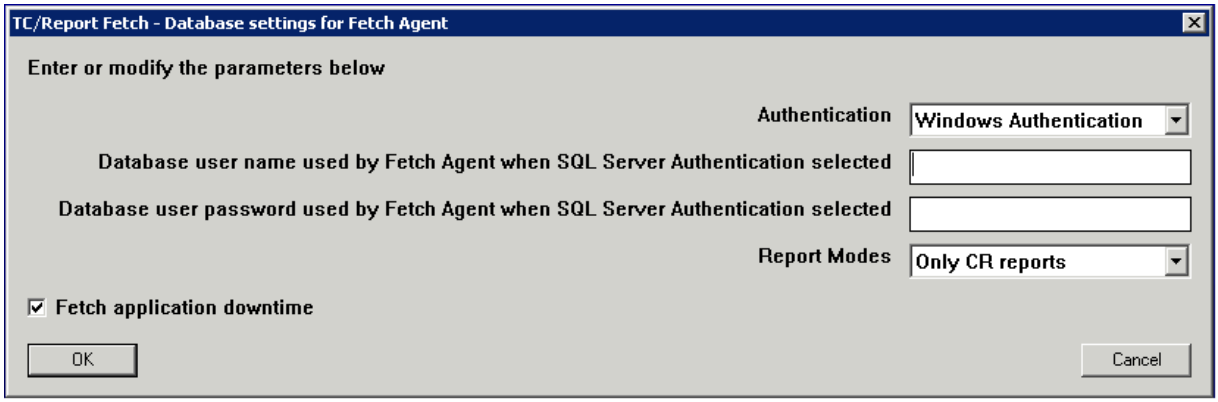
After creating the SQL login and the TC/Report database (optional backup databases can be created later on), you must run Fetch Agent Setup to populate the database with all tables, views and stored procedures that TC/Report needs.

Setup is described in detail later in this manual. This chapter describes a situation where only the TC/Report credentials are available and the database is created by the customer.

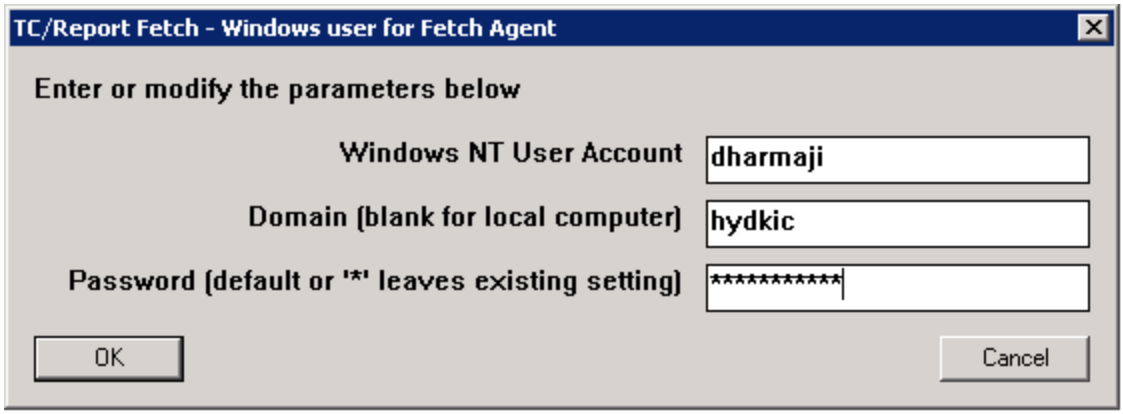
When asked for the database name, specify the name of the new database. Specify the newly created user's credentials for both system user. Use the option "Update database" to let Setup create tables, views and stored procedures in the database.



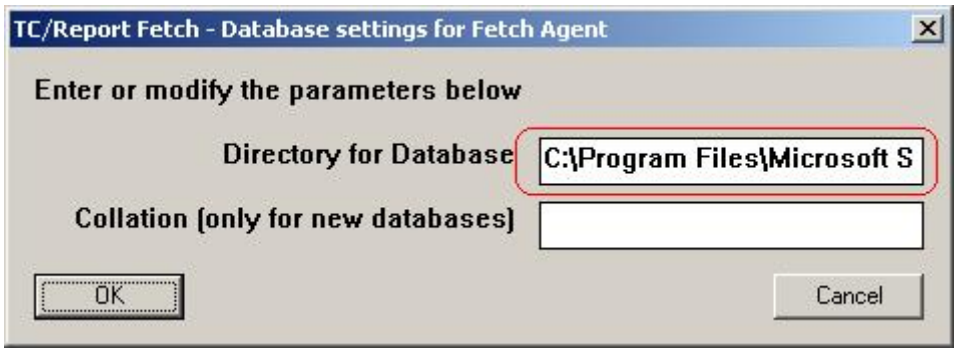
If you want Windows authentication, select **Windows Authentication** in **Authentication** and leave the SQL server authentication fields blank. Else, to use SQL authentication, select **SQL Server Authentication** and specify database user credentials.



For Windows authentication, specify **Windows NT User Account**, **Domain** and **Password** fields for the currently logged in Windows user.



In the Database Settings dialog box, enter the correct database path in the field “Directory for Database”. With SQL Server versions below 2008, the Database Settings dialog box contains a list box labeled “Fetch Agent truncates transaction log”. Set this value to “NO”.



Setup will write warnings into the application event log (event ids 8556 and 8560, as described in section Error Handling), because it cannot give the TCReport user access to the master database. These warnings can be ignored.

Additional Technical Information

The following table is for your information only. SQL Server Administrators might want to know which objects in the Master database are used during TC/Report Setup and at runtime. The following table lists these objects and the reason why they are used. If one of these objects cannot be accessed, a reasonable default handling is used. E.g. if CREATE DATABASE is not allowed, Setup assumes that the database already exists.

Object	Comment
sys.server_principals	Read access, to check if TCReport login exists
sys.database_principals	Read access, to check if TCReport user exists
sys.databases	Read access, to check if TCReport database exists
CREATE DATABASE	To create the TCReport database or a backup database
CREATE PROCEDURE	To create the SP_TC_RECONFIG procedure (used during Setup)

Object	Comment
sp_configure	To read and set the options “show advanced options” and “max server memory”
CREATE LOGIN	To create the TCReport login
CREATE USER	To create the TCReport user
GRANT CREATE DATABASE	To allow the TCReport user creation of backup databases

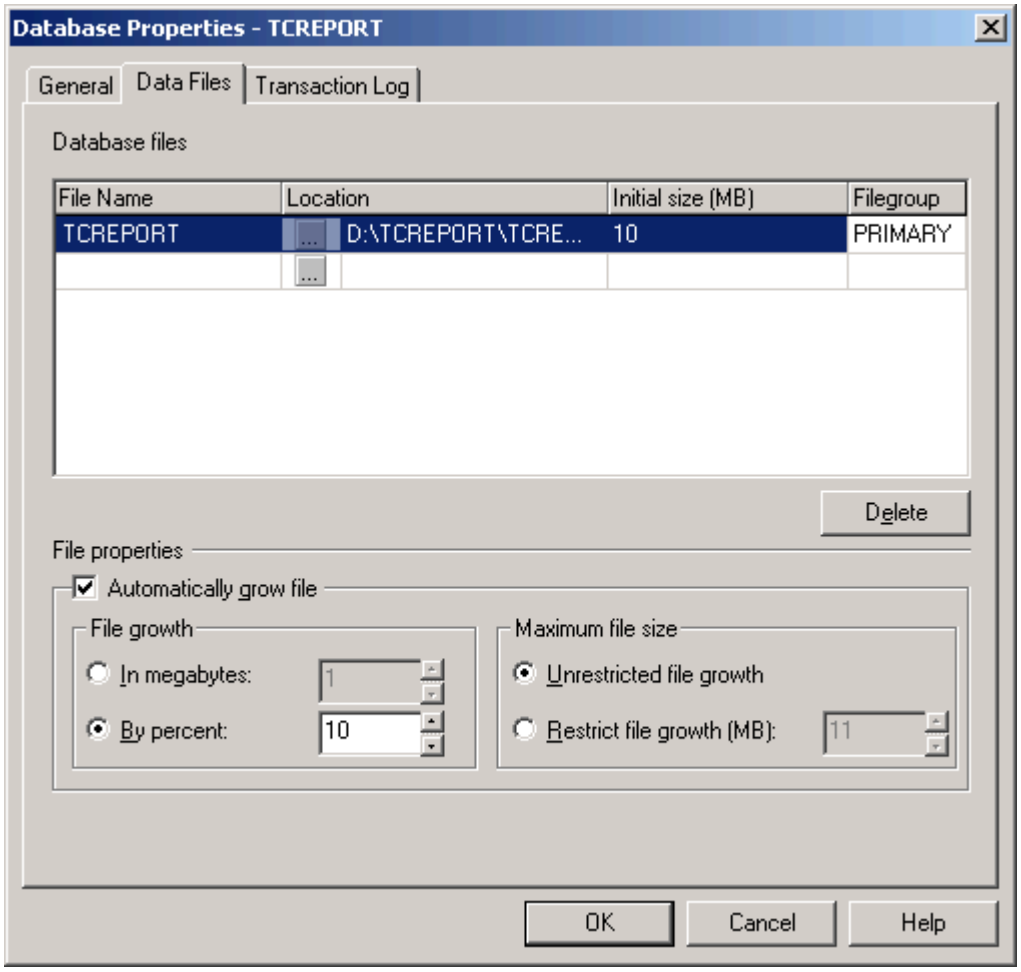
Physical characteristics of the database

The database actually consists of two files, the actual database file and a log file (TCREPORT.MDF and TCREPORT_LOG.LDF), which stores all the database actions. When part of the database is deleted, this is recorded in the log file, which makes the latter grow, so the database will first grow even more before shrinking again beneath the maximum size. That is why the hard disk requirements are double the maximum database size.

KCS Setup creates the TCREPORT database with the following size values:

File Name	Initial size (MB)	Maximum file size	Remark
TCREPORT	10	Unlimited	Data file
TCREPORT_log	10	Unlimited	Transaction Log

The SQL Server Enterprise Manager shows the database properties as follows:



SQL server interprets the initial data file size as a minimum size for this file. This means that it is not possible to shrink the file to a smaller size.

In the same way, it is possible to define a maximum size for the data file, instead of unrestricted growth.

In addition to these file size limits, TC/Report has a maximum size for the database, configurable via the TC/Management Console. If the maximum size of the database is reached, the Fetch Agent will delete the oldest 10% of the table entries and will repeat this until the database size is below its maximum again.

Note You can disable TC/Report's maximum database size check by setting this limit to 0.

To avoid conflicts between the file size limits (that may be changed by an SQL operator) and the database size limit managed by the Fetch Agent, TC/Report ensures that the configured maximum database size (registry: DBMaxSize) is either 0 (disabled) or is within the following limits:

- Lower limit: (sum of database file minimum sizes) * 1.05
- Upper limit (if file growth is restricted for all database files): sum of database file maximum sizes (mdf and ldf files)
- Upper limit (if one or more database files are unlimited in growth): unlimited

TC Management Console reads the limits from the database and adjusts the valid maximum database size that is displayed. If you want to store an invalid value, a message box will pop up, with a text as follows:

Due to database size restrictions, value DBMaxSize is changed from XXX to YYY.

The adapted value (YYY) is stored in the TC/Report configuration.

TC/Report Fetch Agent also reads the limits from the database and adjusts the *DBMaxSize* registry value if necessary. This is done after startup, and then once per hour and whenever the configured *DBMaxSize* limit is reached (before deleting the oldest entries).

If TC/Report Fetch Agent changes the *DBMaxSize* value, it will write the following informational message to the application event log:

ID:8557 Due to database size restrictions, value DBMaxSize is changed from XXX to YYY.

Note With previous versions, TC/Report did not check the size limits of the database files. This could lead to a situation where TC/Report permanently deleted records from a database, in a vain attempt to make a database smaller than its minimum size.

Unfortunately, this mechanism does not handle the overall database size restriction in SQLExpress (4 GB maximum per database).

Content of the Database

Processed data by the Fetch Agents is stored as separate entries in the database.

The main tables in the TC/Report database are:

- Action_table
- Log_table
- Server_table
- Channel_table
- User_table
- Report_table
- Database_table

Note

- The structure of this database might be subject to change in future releases of TC/REPORT.
- This section describes the structure of a database created with KCS Setup 9.2. or above. Databases created with previous KCS Setup versions have varchar columns instead of the nvarchar columns listed here.

Action Table:

Idx	Field Name	SQL Datatype	DB INFO	Description	Comment
1	ID_Action	Int	PRIMARY KEY	Unique ID	
2	Server_ID	nvarchar(30)	INDEX	Unique short name for this TCOSS instance	= Server_ID you specified in TC/ Management Console
3	Direction	Int	INDEX	Direction of the message	These fields are only used by the reports RPTx (and RPTx_GRAPH). The report universal.rpt (Version 1.1) uses the fields MsgClass_Originator and MsgClass_Recipient.
4	Message_Class	nvarchar(30)	INDEX	Class of msg e.g. FAX, TELEX…	
5	Time_Stamp_Local	smalldatetime		Local time of delivery (on KCS)	
6	Time_Stamp_UTC	smalldatetime	INDEX	Coordinated Universal Time of delivery	not used in standard reports
7	File_Name	nvarchar(13)	INDEX	Name of message file	
8	Duration	Int		The duration of sending in seconds	
9	Originator	nvarchar(128)		Originator	
10	Originator_Info	nvarchar(32)		Originator info	
11	Normalized_Orig	nvarchar(128)		Normalized originator	
12	Originator_Group	nvarchar(128)		Originator group	
13	Recipient	nvarchar(128)		Recipient	
14	Recipient_Info	nvarchar(32)		Recipient Info	
15	Recipient_Group	nvarchar(128)		Recipient group	
16	Localized_Addr	nvarchar(128)		Localized address	
17	Normalized_Addr	nvarchar(128)		Normalized address	
18	Cost_Center	nvarchar(128)	INDEX	Cost center	
19	Cost	float		Cost in price units	
20	Size	Int		Message size in bytes	
21	Pages	int		Number of pages	
22	State	int		Status (bit field)	See below for possible values
23	Delay	int		Time in Queue (in seconds)	Time_Action – Time_Created
24	Queue_Length	int		Length of the queue	not supported in this version

Idx	Field Name	SQL Datatype	DB INFO	Description	Comment
25	Attachment_Number	int		Number of Attachments	not supported in this version
26	Attachment_Volume	int		Size of Attachments	not supported in this version
27	Priority	int		Priority of msg, used also for queue	
28	Time_Action	datetime		Time of Delivery	
29	Message_Type	int		Type of msg: Notif(10), Norm(49), Routing(5)	
30	Time_Created	datetime	INDEX	Time of Creation	
31	MsgClass_Originator	nvarchar(30)		Class of msg (FAX, TELEX…) of the originator	
32	MsgClass_Recipient	nvarchar(30)		Class of msg (FAX, TELEX…) of the recipient	
33	Int_Del_Type	int		delivery type (TO_, CC_, BCC)	
34	TS_Channel	nvarchar(30)		Channel. For notification events, this field is filled with the userid of the event owner.	TCSI Version 2.33.00, TCOSS Version 7.38.00
35	TS_Document_Err	nvarchar(30)		error code, which is set when receiving a message	
36	TS_Last_MDA_Action	nvarchar(30)		error code (sending errors)	
37	TS_Env_Name_Posted	nvarchar(30)			
38	Int_Event_Types	int		Event type	See below for possible values
39	Media_Server	nvarchar(30)			TCSI Version 2.33.00, TCOSS Version 7.38.00
40	in_event	int			indicates that an in-event followed this message
41	Int_Er_Recipient	Int		Index of active recipient, for NOTIF report	
42	TS_Last_MDA_note	nvarchar(128)		Error description (sending errors)	
43	TIME_INTENDED	Datetime		Intended time of sending	TCOSS Version 7.41.00 or higher

Idx	Field Name	SQL Datatype	DB INFO	Description	Comment
44	TS_TC_MSG_ID	nvarchar(16)		Unique ID	TCOSS Version 7.41.00 or higher
45	INT_STATE_MASKED	Int		For send retries	TCOSS Version 7.41.00 or higher
46	TIME_REC_END	Datetime		End of reception (incoming fax)	TCOSS Version 7.41.00 or higher
47	TS_REC_CHANNEL	nvarchar(30)		Reception channel	TCOSS Version 7.41.00 or higher
48	TS_REC_SERV_ID	nvarchar(30)		Receiving media server name	TCOSS Version 7.41.00 or higher
49	Distributor	nvarchar(128)		User who distributed message	TCFW version 5.01.01
50	Media_Server_X	nvarchar(30)		Involved media server(for line statistic)	TCOSS Version 7.41.00 or higher
51	Channel_Group_X	nvarchar(2)		Involved channel group (for line statistic)	TCOSS Version 7.41.00 or higher
52	Channel_X	nvarchar(2)		Involved channel (for line statistic	TCOSS Version 7.41.00 or higher
52	Cc_in	nvarchar(10)		Countrycode (in)	
53	Cc_out	nvarchar(10)		Countrycode (out)	
54	ChG_in	nvarchar(2)		Channelgroup (in)	
55	ChG_out	nvarchar(2)		Channelgroup (out)	
56	Subject	nvarchar(128)		Subject	
57	TIME_SCHEDULED	Datetime		planned time of first or next send attempt	TCOSS Version 7.53.00 or higher
58	TIME_SELECTED	Datetime		time when message was selected for sending	TCOSS Version 7.53.00 or higher
59	INT_PAGES_SENT	Int		Number of really transmitted fax pages per send attempt	TCOSS Version 7.85.00 or higher
60	INT_PAGES_SENT_TOTAL	Int		Total number of transmitted fax pages (sum of all send attempts)	TCOSS Version 7.85.00 or higher
61	TS_CORREL_1	nvarchar(128)		Custom field 1	As specified in send order or mapped via SYSCONF line 20.
62	TS_CORREL_2	nvarchar(128)		Custom field 2	As specified in send order or mapped via SYSCONF line 20.
63	TS_CORREL_3	nvarchar(128)		Custom field 3	As specified in send order or mapped via SYSCONF line 20.

Idx	Field Name	SQL Datatype	DB INFO	Description	Comment
64	TS_CORREL_4	nvarchar(128)		Custom field 4	As specified in send order or mapped via SYSCONF line 20.

The Action table is filled by the Fetch Agent and used by the Report Agent to create reports. Most of the parameters gathered from the short-term archive are inserted directly into the Action table. Some parameters are processed before inserting. For processing such parameters, Server and Channel tables are used:

- **Time_Stamp_UTC** is estimated from the Server table by adding the Time_Zone to the Time_Stamp_Local.

  Note The standard reports do not use the Time_Stamp_UTC, because this field does not handle daylight saving times correctly.

- **Message_Class** is obtained from the Channel table. Based on the Server_ID and Channel used for sending and receiving the message, the corresponding Message_Class is inserted into the Action_Table.
- **Other channel specific** are also obtained from the Channel table:
  For correct retrieval of Channel_X, Channel_Group_X, Chg_in, Chg_out, cc_in, cc_out the physical channels must be configured in the Channel_Table.
- **Cc_in, Cc_out**: The countrycodes are retrieved from table Country_Codes.
- **TS_CHANNEL**: For notification events, this field does not contain the channel used for sending the notification. Instead, the field contains the userid of the event owner. In this case, TCOSS does not provide information about the channel used for sending.

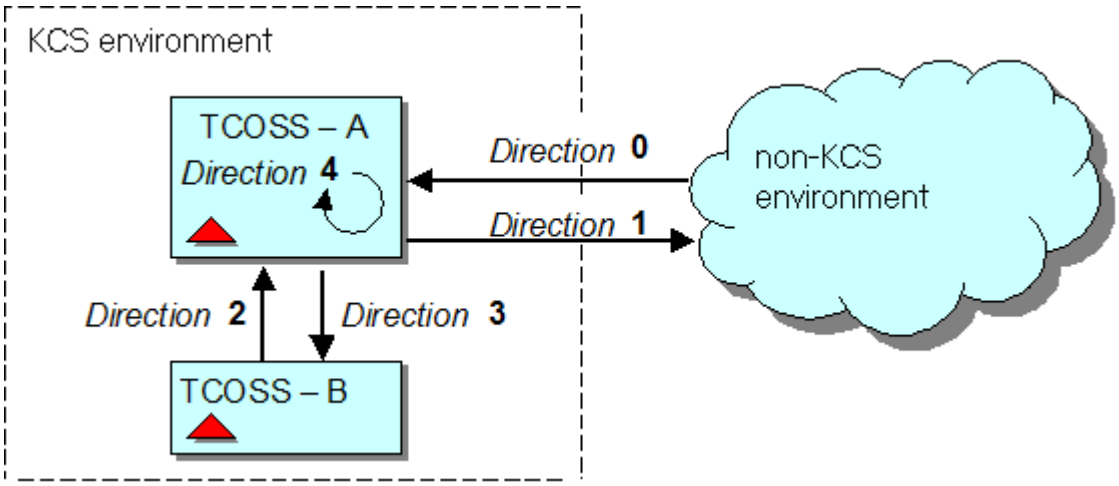
Note There are two fields for the Message Class: MsgClass_Originator and MsgClass_Recipient. MsgClass_Originator is based on the field “Normalized_Orig” (TCSI field TS_NORMALIZED_ORIG). MsgClass_Recipient is based on the field “Localized_Addr” (TCSI field TS_LOCALIZED_ADDR). It is possible that both fields contain a value, e.g. MsgClass_Originator = “FAX”, MsgClass_Recipient = “PRINT”.

The first version of TC/Report (Release 1.00.xx) only had one field called “Message_Class”. The only way to find out if the message class was based on the originator or on the recipient was the field “Direction”. These fields (Message_Class and Direction) are not necessary in the current TC/Report version.

Direction field identifies the direction of the message. It can contain following numbers:

0 – incoming message sent from non-KCS environment
1 – outgoing message sent to non-KCS environment
2 – incoming message sent from different TCOSS server
3 – outgoing message sent to different TCOSS server
4 – message sent/received inside one TCOSS server

Note In the new reports delivered with this release, the Direction field is not used any more.



Message state overview:

The TCSI field INT_STATE (stored in column State of the Action_Table) holds a numeric value defining the message state and a bitfield with flags. The numeric value consists of the 10 least significant bits, all other bits are flags.

State without flags	Value	Description
ST_POSTED	150	new entry put into mail system
ST_WAIT_CONV	160	waiting for document conversion
ST_CONVERTING	170	document conversion in progress
ST_CONV_END	180	document conversion complete
ST_WAIT_SEND	190	put into in-box of gateway user or public line channel
ST_WAIT_RETRY	200	wait after unsuccessful send attempt
ST_SENDING	300	sending (on public line channel or to gateway user)
ST_BREAK0	310	sending ended with BREAK=0 (checkpoint)
ST_BREAK1	320	sending ended with BREAK=1 (checkpoint)
ST_BREAK2	330	sending ended with BREAK=2 (checkpoint)
ST_BREAK3	340	sending ended with BREAK=3 (checkpoint)
ST_BREAK4	350	sending ended with BREAK=4 (checkpoint)
ST_BREAK5	360	sending ended with BREAK=5 (checkpoint)
ST_CANCELLED	400	cancelled by originator
ST_CONV_FAILED	405	document conversion failed
ST_TIMEOUT	410	latest delivery time limit expired
ST_END_OF_RETRIES	420	no more send attempts
ST_REJECTED	430	rejected by authorizer
ST_CORRECTED	450	corrected with posting-for-correction

State without flags	Value	Description
ST_SENT	500	sent to next node, not yet delivered
ST_DISTRIBUTED	600	distributed with posting-for-correction
ST_ROUTED	610	transferred to next TCOSS node, terminated
ST_DELIVERED = ST_RECEIVED	650	delivered to recipient's mail box, archivation of in-box entries
ST_UNREAD	700	new entry in in-box
ST_AUTO_FORWARDED 1)	710	automatically forwarded (unread)
ST_READ	800	read by recipient
ST_REPLIED 1)	810	replied by recipient
ST_PROGRESSED 1)	820	progressed by recipient

1) State value for future extension, not used in current implementation

State values ST_DELIVERED or higher may be combined with flag ST_TERMINATE (2048 decimal). ST_TERMINATE indicates that this state update is the final one.

Event types overview:

The value of the Int_Event_Types column from the Action_Table is defined through several flags:

State	Value	Description
IN_MAIL	1	Normal message
DEL_NOTIF	2	Delivery notification
NON_DEL_NOTIF	4	Non-delivery notification
BACK_REC	8	Send copy
MWON	16	MWON event
MWOFF	32	MWOFF event
ALERTS	64	Alert message
NUMBER_LOCKING	512	Not relevant in this context
GATEWAY	1024	Not relevant in this context
FROM_ACTION	2048	Created by an event
NO_IN_ARC	4096	Not relevant in this context
AUTO_REJECT	8192	Not relevant in this context
ARC_ALL_ATTEMPTS	16384	Not relevant in this context
IN_RELEASE	65536	Incoming release event
OUT_RELEASE	131072	Outgoing release event

For example, a message which was triggered by an IN-event would have IN_MAIL (1) + FROM_ACTION (2048) = 2049 in this column.

View Action_Table_Descr

This is a view of the Action_Table with additional fields for message state, delivery type etc. These are the additional columns:

Column Name	Possible values
TXT_PRIORITY	LOW, NORMAL, HIGH
TXT_MESSAGETYPE	MESSAGE,NOTIF,ROUTING
TXT_DELTYPE	TO,CC,BCC,AUTH
TXT_STATE	SUCCESS,FAILURE
TXT_RETRIES	WAIT_RETRY,NO RETRY
IsEvent	1,0

Log Table:

Idx	Field Name	SQL datatype	DB INFO	Description
1	ID	Int	PRIMARY KEY	Unique ID
2	Server_Id	nvarchar(30)	INDEX	Unique short name for this TCOSS instance
3	Time_Stamp_Local	Datetime	INDEX	Local TCOSS time when log entry was created
4	Log_ID	Varchar(128)	INDEX	Log type
5	ObjectName	nvarchar(128)	INDEX	Object to which the log entry applies
6	ValueType	Int		0(integer) or 1(string)
7	ValueString	nvarchar(128)		String value (for ValueType=1)
8	ValueInt	Int		Integer value (for ValueType=0)
9	Time_Stamp_UTC	Datetime	INDEX	UTC time when log entry was created
10	Msg_ID	Int	INDEX	ID for complete log message
11	First_ID	Int		unique for every TCSI log entry
12	Log_User	nvarchar(128)		originator of log entry

The Log table is filled by the Fetch Agent and used by the Report Agent to create the QUEUELEN report. It holds information from the log entries created by TCOSS version 7.41.00 or higher.

Note You need TCfW version 5.01.01 for configuring log entries.

Server Table:

Idx	Field Name	SQL Datatype	DB INFO	Description
1	Server_ID	nvarchar(30)	PRIMARY KEY	Unique short name for this TCOSS instance
2	Server_Description	nvarchar(80)	INDEX	Descriptive name (used in the reports)
3	Node	char(1)		Node character
4	Time_Zone	Int		TCOSS time zone

Idx	Field Name	SQL Datatype	DB INFO	Description
5	Start_Entry	varchar(50)		start entry
6	Server_Path	nvarchar(128)		TCOSS path
7	User_ID	nvarchar(128)		User ID for login to KCS
8	User_Password	nvarchar(128)		Password (Encrypted)
9	channel_changed	Int		temporary field for the configuration
10	currency	nchar(3)		
11	disable_flag	Bit		
12	last_entry	Datetime		When fetch agent is still fetching, last_entry is the Time_Action field of the real last entry. If there are no more entries, the last_entry field gets the actual date / time of the TC/Report machine to make it possible to create reports until now.
13	CustomerName	nvarchar(180)		Name of customer (for ASPs)
14	FetchLogEntries	Bit		1 if log entries shall be fetched
15	FetchSendAttempts	Bit		1 if send attempts shall be fetched
16	Fetch_Subjects	Bit		1 if Subjects shall be fetched
17	Fetch_Users	Bit		1 if user information shall be fetched
18	dt_fullfetch	Datetime	Used internally by Fetch Agent	Date and time of last full fetch
19	Is_Media_Server	Bit	should be 1 for media servers	If true, this server does not show up in report headers
20	ProbeUser_ID	nvarchar(128)		Probe user (for TC/Probe)
21	ProbeUser_Password	nvarchar(128)		Probe user password
22	ProbeAgent_Group	nvarchar(30)		Probe agent group responsible for this server
23	NodeAlias	nvarchar(128)	For TC/MA	Used in LCR systems only. User names, channels and channel groups are here prefixed with the NodeAlias. TC/Report must recognize and correctly interpret this prefix If server is an LCR node: Node alias from rr99, without trailing – character..
24	Fetch_Events	Bit	For TC/MA	With this flag, you configure whether media server events are assigned to the media server (needs a separate tracking agent) or to the storage server. 1: create events for this server (if configured globally) 0: do not create events for this server

Idx	Field Name	SQL Datatype	DB INFO	Description
25	CustomerIdField	Integer	For TC/MA	Specifies which user profile attribute holds the customer id (if single TCOSS instance is shared by several customers). This is a TCSI field ID. Default is 0. Possible values: 0: only 1 customer 191: costcenter 201: group 205: company 206: department 207: fullname 208: salutation 209: freetext 282: location
26	Last_Change_RS	nvarchar (268)		Starting point for fetching recipient store changes.
27	Last_Change_US	nvarchar (24)		Starting point for fetching user store changes.

The Server table is used for translating the unique Server ID to a printable server description during creating reports via Crystal Reports (only Server ID is used during processing and finally the Server Description is inserted in the report).

Channel Table:

Idx	Field Name	SQL Datatype	DB INFO	Description
1	ID_Channel	int	PRIMARY KEY	Unique ID
2	Server_ID	nvarchar (30)	INDEX	TCOSS instance name
3	Channel	nvarchar (128)	INDEX	Channel e.g. “F”
4	Message_Class	nvarchar (30)		Message class e.g. “FAX”

The Channel table is used by the Fetch Agent to insert the Message_Class parameter and to retrieve information about for physical channels (fax or telex channels / channel groups).

User Table:

This table holds a history of user profile changes. There may be several entries for a user, but only one of them is valid at a time.

Idx	Field Name	SQL Datatype	DB INFO	Description
1	ID_User	Int	PRIMARY KEY	Unique ID
2	Server_Id	nvarchar(30)	INDEX	TCOSS instance name
3	User_Id	nvarchar(128)	INDEX	KCS User ID

Idx	Field Name	SQL Datatype	DB INFO	Description
4	Department	nvarchar(80)		Department
5	Company	nvarchar(80)		Company
6	Fullname	nvarchar(80)		Fullname
7	FreeTextFld	nvarchar(80)		Freetext
8	Salutation	nvarchar(32)		Salutation
9	Mailsystem	Integer		Mailsystem (“user belongs to”) is translated to text constants in TCLINESN report
10	DateValid	Datetime		Start of validity
11	DateInvalid	Datetime		End of validity
12	QueueUser	Int	For TC/MA integration	All users with “visible in outbox” flag and valid media type are considered as queue users. ++INVALID and other users configured for automatic reject of messages are also regarded as queue users. Purpose: distinguish between user mailboxes and queues that are used by applications 0: normal user 1: queue user
13	CostCenter	nvarchar(12)	For TC/MA integration	Cost center From user store entry field TS_COST_CENTER
14	UserGroup	nvarchar(127)	For TC/MA integration	Group From user store entry field TS_GROUP
15	Location	nvarchar(8)	For TC/MA integration	Location From user store entry field TS_LOCATION
16	MediaType	Integer	For TC/MA integration	Media type From new user store entry map field MessagingAnalyser.MediaType Fixed constants + several custom constants
17	ApplicationName	nvarchar(30)	For TC/MA integration	Application Name From new user store entry map field MessagingAnalyser.Application
18	Suspect_Invalid	Bit		Used internally for detection of deleted users during full user import

Fields used only for TC/MA integration:

The purpose of the new fields **CostCenter**, **UserGroup** and **Location** is to offer reports that are grouped or filtered by the user’s cost center, group and location.

The new fields **QueueUser**, **MediaType** and **ApplicationName** allow to distinguish queues from real users, and to assign a media type (e.g. sms, internet mail etc.) to messages sent via queues.

Additionally, queues can be assigned an application name, e.g. all queues polled by parallel instances of TC/LINK-MX might have application name TCLINKMX. This allows filtering for messages that were processed by certain applications.

The fields **MediaType** and **ApplicationName** can be edited via TCFW.

TCLINK.EXE 2.08.02 and above sets default values for these fields, corresponding to the link type, when creating new queue users.

Backup implications:

When the Fetch Agent creates a backup, it populates the User_Table in the backup database with all user records that are valid for the time period covered by the backup.

Size limit implications:

When checking maximum database size, the Fetch Agent also counts the size of the User_Table. If old records are removed from the Action_Table and Log_Table, the Fetch Agent deletes all user records that apply only to the removed send orders.

Report Table:

Idx	Field Name	SQL Datatype	DB INFO	Description
1	reportname	nvarchar(128)	PRIMARY KEY	the report filename (without .rpt)
2	checkcurrency	bit		= 0 if the currency should not be checked, when this report is requested
3	checkdate	bit		This setting decides when a report requested with “wait=YES” shall be sent. = 1: report is sent when the end date has been fetched = 0: report is sent when the local time on the Report Agent computer has reached the end date. Note: Depending on the report parameter range_check, additional checks (is start date also in the database) will be done..

When creating a report for several KCS servers, the Report Agent checks if the same currency is configured for all of them and refuses to create a report if there are different currencies. You can disable this check per report type by entering reports manually into this table, and setting checkcurrency to 0.

In a default installation, currency check is disabled for the following report types:

NOTIF
TCJOB, TCDL
DIST, DIST_GRAPH
QUEUELEN, QUEUELEN_GRAPH
LINES, LINES_GRAPH
TCLINESN, TCLINESN_GRAPH
TCPROBE, TCPROBED
TCWAIT, TCWAIT_GRAPH
TCAPP

The Report Agent also checks if the specified report interval is already fetched in the Action_Table. This check is not necessary and therefore disabled for the following report types:

TCPROBE, TCPROBED, NOTIF, TCAPP

Database Table

Idx	Field Name	SQL Datatype	DB INFO	Description
1	DB_ID	Int	PRIMARY KEY	Unique ID
2	filename	Nvarchar (128)		Filename of the Backup database (without .mdf)
3	datefrom	datetime		first entry of the database
4	dateto	datetime		end of the time period of the database
5	ready	Bit		= 1, if the fetch agent doesn't store entries on this database anymore
6	deleted	Bit		database is already overwritten, because the maximum number of databases is limited
7	StartOfPeriod	Datetime		For periodic backup: start of current backup period

The Database table is used for the automatic backup creation.

Country Codes Table

Idx	Field Name	SQL Datatype	DB INFO	Description
1	Cc	Varchar(10)		Country code (dial prefix)
2	Description	Varchar(80)		Description (not used)
3	Cc_class	Varchar(1)		F for Fax, X for Telex

The Country Codes table is used for reports grouped by country code.

The following tables are of minor importance:

Report_Errors Table

The Report Agent uses an additional table to track which errors need to be notified to the user. This table is maintained by the Report Agent only. Do not change it!

Idx	Field Name	SQL Datatype	DB INFO	Description
1	RE_ServerID	nvarchar(128)		Server ID
2	RE_FileName	nvarchar(13)		Message filename
3	RE_CreatedAt	Datetime		Message creation time
4	RE_ErrNumber	Int		Error number

Idx	Field Name	SQL Datatype	DB INFO	Description
5	RE_Checked	Bit		Flag if error message was sent

NotifTmp Table

This table is used for creation of the NOTIF report. It is populated by the Report Agent before the report is created.

Idx	Field Name	SQL Datatype	DB INFO	Description
1	File_Name	nvarchar(13)		Message filename
2	Duration	Int		Duration
3	Recipient_Info	nvarchar(13)		Recipient Info
4	State	Int		Message status
5	Delay	Int		Delay
6	Time_Created	Datetime		Message creation time
7	INT_DEL_TYPE	Int		Delivery type
8	TS_LAST_MDA_ACTION	nvarchar(30)		Error code
9	TS_LAST_MDA_NOTE	nvarchar(30)		Error text
10	NumAttempts	Int		Number of send attempts
11	PCName	nvarchar(255)		Report agent computer name
12	InstanceName	nvarchar(255)		Instance of report agent

View Action_User

This view is used internally only. It contains all columns view Action_Table_Descr and the following additional columns:

Column Name	Description
RecipDept	Recipient Department
RecipCompany	Recipient Company
RecipFullname	Recipient Fullname
RecipFreetext	Recipient Freetext
RecipSalutation	Recipient Salutation
RecipMailsystem	Recipient Mailsystem
OrigDept	Originator Department
OrigCompany	Originator Company
OrigFullname	Originator Fullname
OrigFreetext	Originator Freetext
OrigSalutation	Originator Salutation
OrigMailsystem	Originator Mailsystem

Column Name	Description
FromServer	Server or media server for incoming fax or telex
FromChannel	Channel for incoming fax or telex
ToServer	Server or media server for outgoing fax or telex
ToChannel	Channel for outgoing fax or telex

View AU1

This view is used internally only. It contains a subset of the Action_User entries (only send orders from or to real fax or telex channels).

View AU2

This view is the basis for the TCLINESN report. It contains selected columns from the AU1 view and additionally the following columns:

Column Name	Description
channel_out	hardware channel used for sending
channel_in	hardware channel used for receiving
channel_out_intended	channel or channel group from recipient address

The field channel_out_intended is used to recognize cancelled and timed out send orders (in these cases TS_CHANNEL is no hardware channel). View AU2 contains only sendorders that were sent to / received from fax or telex channels.

This view gets the channel information directly from the channel_table. This means, that you can correct the channel configuration via TC Management Console even after the records were fetched to the database, - and no update of the action_table is necessary. On the other hand, this means that the TCLINESN report uses only the latest information: If a channel number is used for fax at the time of report creation, it is interpreted as a fax channel for all send orders that are covered by the report.

View JOBUSER

This view is used internally by the TCDL report. It correlates Multistream jobs and job originator.

Column Name	Description
JOB_SERVER	TCOSS instance name
JOB_ID	Job id
JOB_MSG_ID	TS_TC_MSG_ID of the job
JOB_USER	Originator of the job
Time_Stamp_Local	Time of job start event (local time)
Time_Stamp_UTC	Time of job start event (UTC time)

View JOBDL

This view is used internally by the TCDL report. It correlates Multistream jobs and distribution lists.

Column Name	Description
JOB_SERVER	TCOSS instance name
JOB_ID	Job id
JOB_MSG_ID	TS_TC_MSG_ID of the job
JOB_DL	1 distribution list of the job
Time_Stamp_Local	Time of job start event (local time)
Time_Stamp_UTC	Time of job start event (UTC time)

View VIEW_JOB_DL

This is the base view for the TCDL report.

Column Name	Description
Time_Stamp_Local	Time of job start event (local time)
Time_Stamp_UTC	Time of job start event (UTC time)
JOB_SERVER	TCOSS instance name
JOB_ID	Job id
JOB_DL	1 distribution list of the job
JOB_USER	Originator

View JOBACTION

This view is used internally by the TCJOB report. It provides a correlation between a job and its send orders (sum of duration, pages in, pages out).

Column Name	Description
JOB_SERVER	TCOSS instance name
Time_Stamp_Local	Time of job start event (local time)
Time_Stamp_UTC	Time of job start event (UTC time)
JOB_USER	originator
JOB_ID	Job id
JOB_MSG_ID	TS_TC_MSG_ID of the job
PagesIn	Number of incoming pages
Duration	Sum of duration of resulting send orders
PagesOut	Number of pages successfully sent out (incoming pages * number of successful send orders)

View VIEW_JOB_ACTION

This is the base view for TCJOB report. It correlates jobs, send orders and distribution lists. There is 1 record per distribution list and job, e.g. for a job with 9 distribution lists there will be 9 records.

Column Name	Description
JOB_SERVER	TCOSS instance name
Time_Stamp_Local	Send time of single message (local time)
Time_Stamp_UTC	Send time of single message (UTC time)
Duration	Duration of single message
JOB_USER	Job originator
PagesIn	Number of incoming pages
PagesOut	Number of pages successfully sent out (incoming pages * number of successful send orders)
JOB_ID	Job id
JOB_DL	Distribution list number

Tables used for application downtime monitoring

The **Workstation_Table** holds information about single workstations:

Field Name	SQL Datatype	DB INFO	Description
Workstation	nvarchar(30)	Unique	Name of the workstation
FetchTime	Varchar(5)		Fetch time in format hh:mm If empty, fetching of event log entries is disabled
NextFetch	Datetime		Date and time when next fetch is due
LastID	Varchar(255)		ID of last fetched event log entry
wmiDomain	nvarchar(255)		Domain of account used for event log access
wmiUsername	nvarchar(255)		Name of account used for event log access
wmiPassword	nvarchar(255)		Password of account used for event log access, stored encrypted
wmiNamespace	nvarchar(255)		Namespace used of event log access, typically empty, - in nonstandard WMI installations it must be set to *root\cimv2*

The **AppGroup_Table** holds information about single application groups:

Field Name	SQL Datatype	DB INFO	Description
ID	Int	Identity	Unique ID
GroupID	nvarchar(30)		Group name
Customer	nvarchar(30)	Index	Customer whom the group belongs
Description	nvarchar(128)		Group description

The **Application_Table** holds information about single applications:

Field Name	SQL Datatype	DB INFO	Description
ID	Int	Identity	Unique ID
AppID	nvarchar(30)		Application name
Workstation	nvarchar(30)	index	Workstation where the application runs
RegKey	nvarchar(30)	index	Registry subkey of the application
AppType	Varchar(30)		Application type

The **App_AppGroup_Table** holds the correlation between application groups and applications.

Field Name	SQL Datatype	DB INFO	Description
ID	Int	Identity	Unique ID
GroupID	Int	Index	Application Group ID
AppID	Int	index	Application ID

The **App_DownTime_Table** holds information about the time intervals when single applications were not available. This table is the basis for the TCAPP report.

Field Name	SQL Datatype	DB INFO	Description
ID	Int	Identity	Unique ID
CustomerID	nvarchar(30)	Index	Customer name
AppGroupID	nvarchar(30)	Index	Application group name
Workstation	nvarchar(30)	index	Workstation name
AppID	nvarchar(30)	Index	Application name
ApplicationID	Int		Unique ID of the application
DownFrom	Datetime		Start of downtime
DownTill	Datetime		End of downtime (NULL if the application is still down)

The table **RegKeys2TCApplication** holds meta-information about KCS applications. This table is consulted for automatic recognition of applications and for evaluation of event log entries. This table is deleted and created newly at every Setup, in order to keep the application-specific information up-to-date.

Note Events logged by TCSRV need not be specified here, - they are evaluated automatically (e.g. 21506 for “started”, 21527 for “stopped”).

Field Name	SQL Datatype	DB INFO	Description
ID	Int	Identity	Unique ID

Field Name	SQL Datatype	DB INFO	Description
SubKey	Varchar(255)		For automatic recognition of applications, specifies a registry key that can be used to identify the application
RegValue	Varchar(30)		For automatic recognition of applications, specifies a registry value that can be used to identify the application
RegData	Varchar(50)		For automatic recognition of applications, specifies data in the above registry value that can be used to identify the application
AppType	Varchar(30)		Application type (see table above)
TCOSSPath	Varchar(255)		Registry value holding the TCOSS server path (if existing for this application)
Customer	Varchar(255)		Registry value holding the customer name (if existing for this application)
StartEventIDs	Varchar(255)		Comma separated list of event IDs that signal that the application has become available
StopEventIDs	Varchar(255)		Comma separated list of event IDs that signal that the application has become unavailable

Other tables and views used internally

The following objects are created by TCReport during runtime. They are not described here in detail.

- table “message_table” (for the Notif report)
- view “bothtables” (for deletion of oldest entries)

Other tables and views used by KCS Probe Agent

The product KCS Probe Agent shares the TC/Report database. The database components used by this product are described in a separate document, - the TCPROBE Manual.

Other tables and views used by KCS Messaging Analyser

The product KCS Messaging Analyser (TC/MA) shares the TC/Report database. The database components used by this product are described in a separate document, - the TC/MA Manual.

Chapter 4

Installation

In KCS Setup, both agents can be installed separately. There must be 1 Fetch Agent per database. The number of Report Agent instances can vary according to the customer's needs. For example, an ASP customer might have one Report Agent per client.

Additionally, Setup allows you to install a TC Management Console applet for creating report requests.

The TC/Report database can only be installed on MS-SQL Server (versions see Minimum Requirements) installed.

The agents can run on any computer in the network (not necessarily on the SQL server) and can be installed separately.

You need one **TC/Report** license which includes operation of Fetch Agent and Report Agent(s).You can install multiple Report Agents on different computers (it is for free).

Report Agent Setup installs the Crystal Reports 11.5 runtime library on the local computer.

The following users/rights are needed:

Fetch Agent:

- Dedicated Windows user (member of **local Administrators**, right to **log on as a batch job**) or system account.
- TCOSS user – see Step 2, (one per server, right to list in/outbox of all users, configured in the TC Management Console).

  For fetching send attempts or log entries, and for fetching message subjects, the user must have the right to OPEN all in/outboxes.

  For fetching user information, the user must have the right to read System Address Book and User Profiles.
- Database user (full rights, only used by TC/Report, configured during setup).

  Note Do NOT use the sa user!

Report Agent:

- Windows user (network access if direct PRINT or FILE destinations are used, configured during setup) – the Windows user must be a **local Administrator** and needs the right **log on as a batch job** (Windows **User manager**/ **policies**/ **user-rights**/ **show advanced user rights**).
- TCOSS user – see Step 2, create TC/Report user (one server only, standard user, needs address, read access to user profiles and address book, set gateway user (visible in outbox is active), no events recommended, configured during setup).

- Database user (full rights, only used by TC/Report, created during Fetch agent setup). Must be **same account as used by the Fetch agent**.

TCOSS User that requests reports:

- **Reporting** right or (for backward compatibility) **List in/outbox of all user profiles** right required.

Step 1 – Installation of the Database

The database server can be on any computer in the LAN. You can either use SQL Server or SQL Server Express. For supported versions, see *Minimum Requirements*.

Installation of MS SQL Server

Installation steps on the MS SQL Server

MS SQL Server can be installed on any computer in the network. For use with TC/Report, the SQL Server must be configured for mixed authentication mode.

Installation steps on the TC/Report computer

For Fetch and Report agent, Microsoft ADO 2.5 or above is needed. ADO is part of the Windows Server operating systems.

Note If for any reason ADO is not installed on the workstation, you can install it by running mdac_typ.exe from the KCS setup CD. This will install ADO 2.8.

For integration with SQL Server 2005, ADO 2.8 SP1 or a higher version is the needed. This version can be downloaded from the Microsoft web page.

If you plan to use one of the batch jobs **timed_backup** or **timed_cleanup**, or any of the SQL server command line tools like **sqlcmd**, you need the MS SQL client components. They can be installed from the SQL Server setup disk.

Choose **Client Connectivity only** in Setup. With SQL Server 2005, you can also run SQLREDIS.EXE from the SQL Server CD (directory Ent\X86\Other).
TCP/IP is one of the default protocols; therefore no protocol change is needed.

Optional: Change Password of sa User

SQL Server Setup creates a user “sa” who has full permissions on all databases. With SQL Server, this user needs a “strong password”.

The requirement for a strong password can be disabled via the tool sqlcmd.

How to disable strong password policy:

```
sqlcmd -U sa -P Pa$$w0rd
1>alter login sa with check_policy=off
2>go
```

How to change the password:

You can use “alter login” also to configure a password for this user:

```
sqlcmd -U sa -P Pa$$w0rd
1> alter login sa with password = 'newPassword'
2>go
```

Type “EXIT” to exit the tool sqlcmd.

Step 2 – Create KCS User Accounts

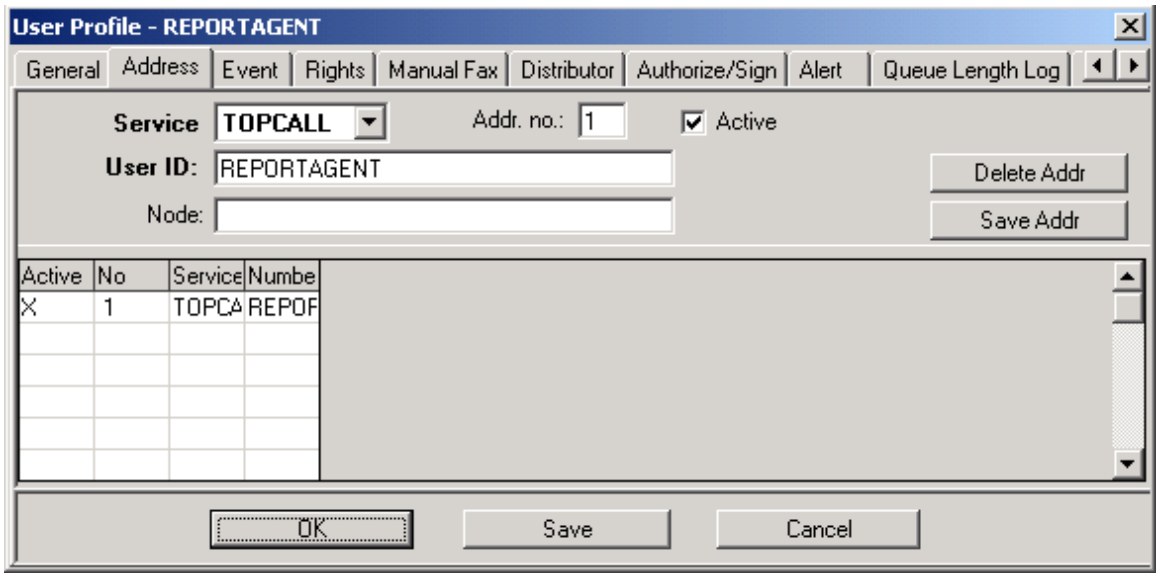
Both the Fetch Agent and the Report Agent log on to the Kofax Communication Server via a KCS user account. You can use one user for both agents, provided that you grant him the user rights needed by both agents.

If there are several Report Agents, each of them must have a different KCS user. Otherwise, duplicate reports can be created.

Create the user(s) with TCfW.

(Refer to the KCS Client Applications Administrator Manual and to the TCfW User Manual.)

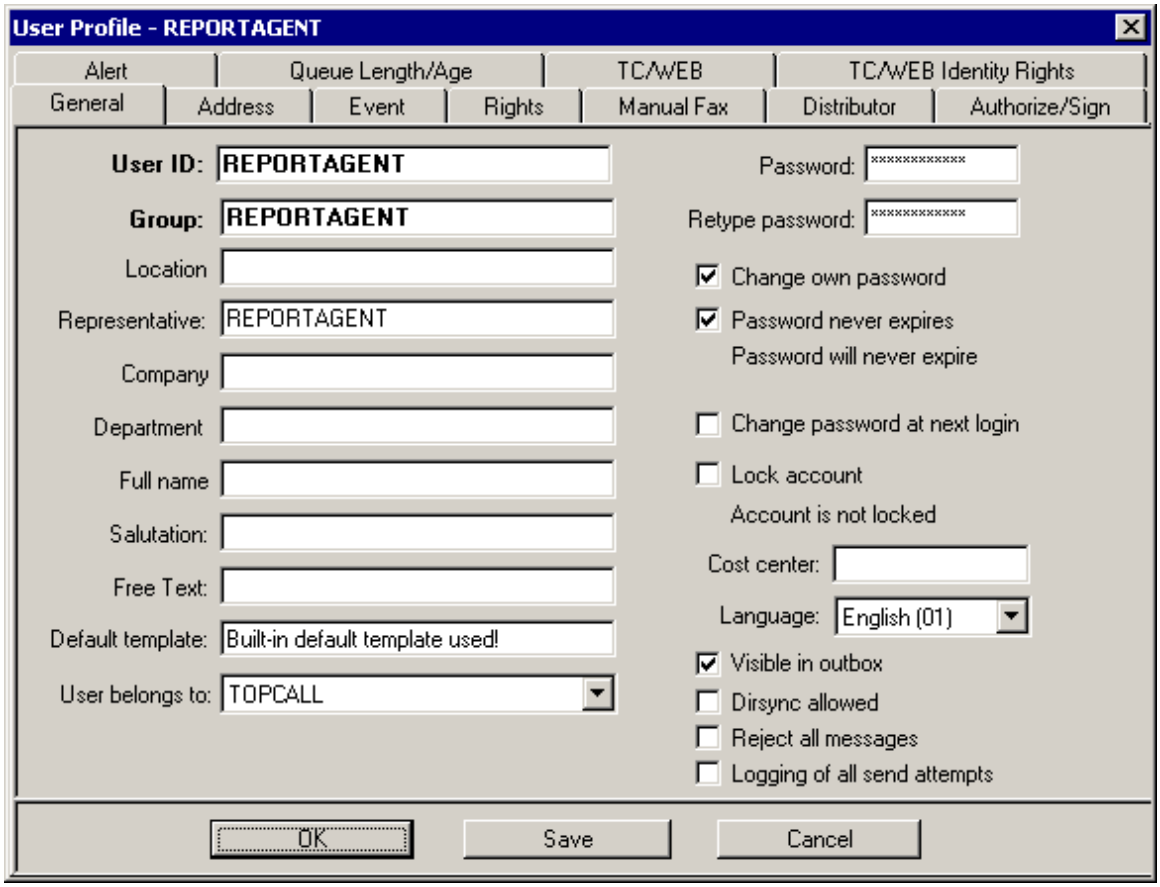
The user needs a KCS internal address (service “TOPCALL”):

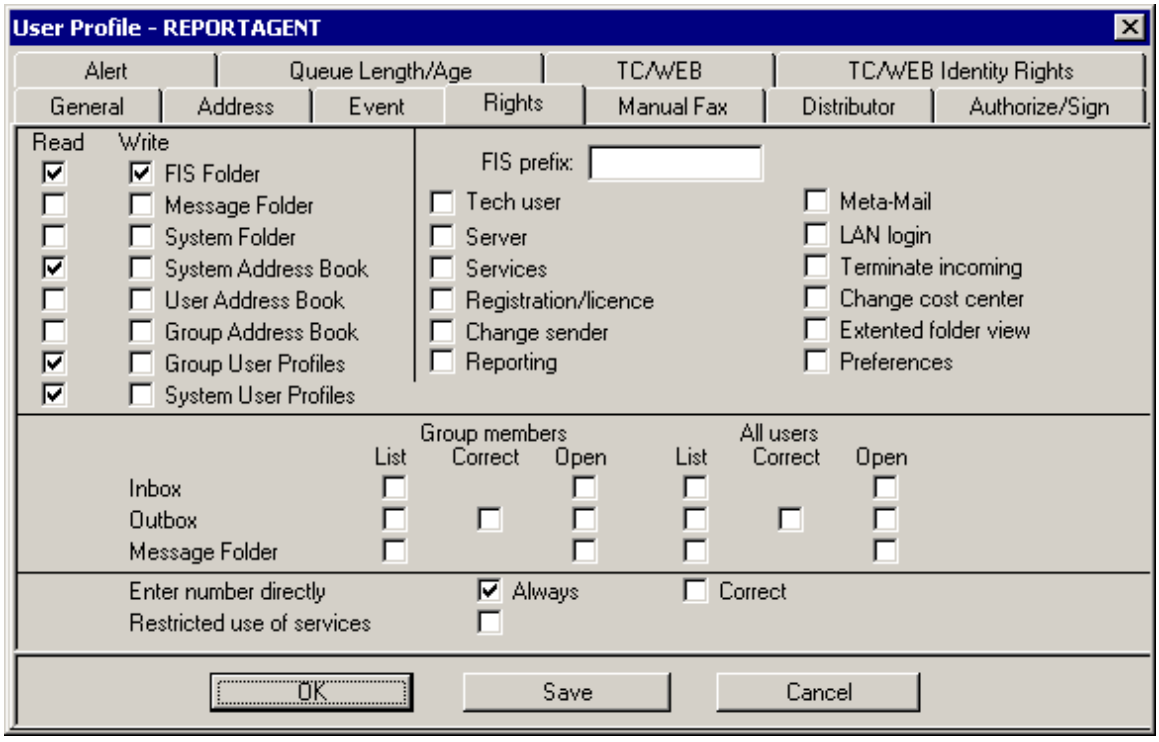


For the Report Agent:

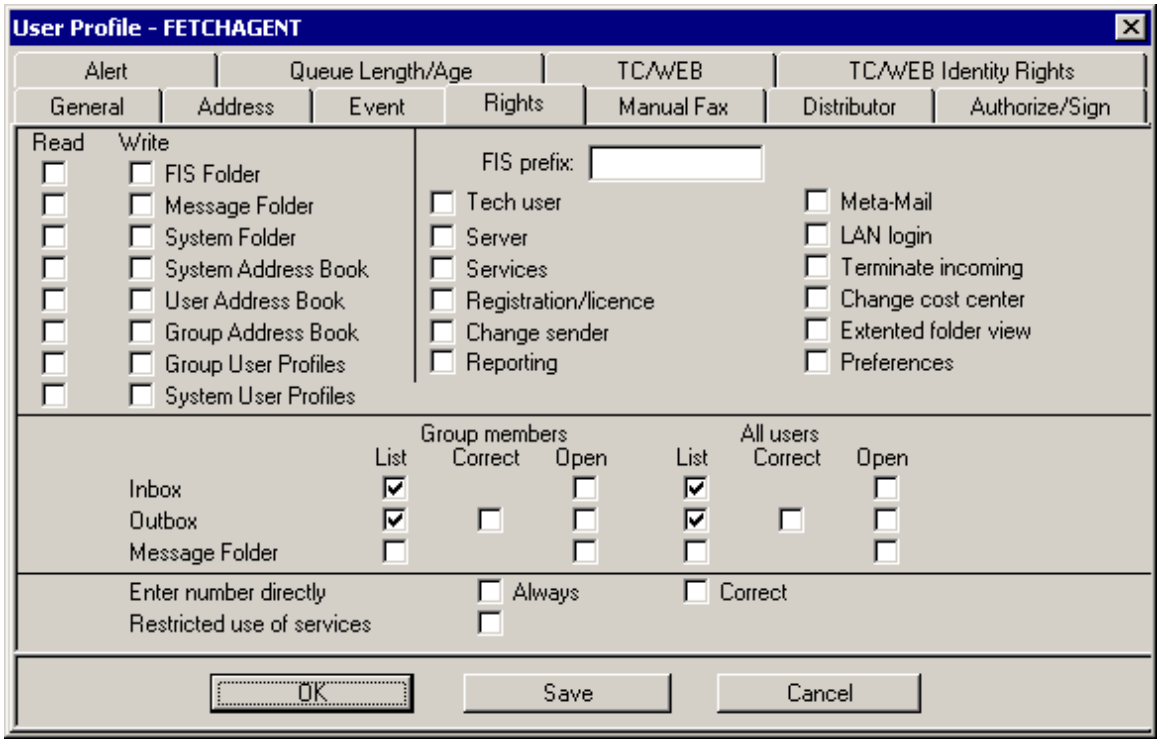
This KCS user will be used to accept command messages sent to the Report agent.
The account needs the following rights and settings:

- Set gateway user (**Visible in outbox** checkbox is active)
- **Read Group and System User Profiles** and
- **Write FIS Folder** (if report configuration files shall be stored in FIS folder)
- **Read System Address Book**


User Profile - REPORTAGENT
Alert
Queue Length/Age
TC/WEB
TC/WEB Identity Rights
General
Address
Event
Rights
Manual Fax
Distributor
Authorize/Sign
User ID:
REPORTAGENT
Group:
REPORTAGENT
Location
Representative:
REPORTAGENT
Company
Department
Full name
Salutation:
Free Text:
Default template:
Built-in default template used!
User belongs to:
TOPCALL
Password:
Retype password:
Change own password
Password never expires
Password will never expire
Change password at next login
Lock account
Account is not locked
Cost center:
Language:
English (01)
Visible in outbox
Dirsync allowed
Reject all messages
Logging of all send attempts
OK
Save
Cancel



Permissions for the Fetch Agent:

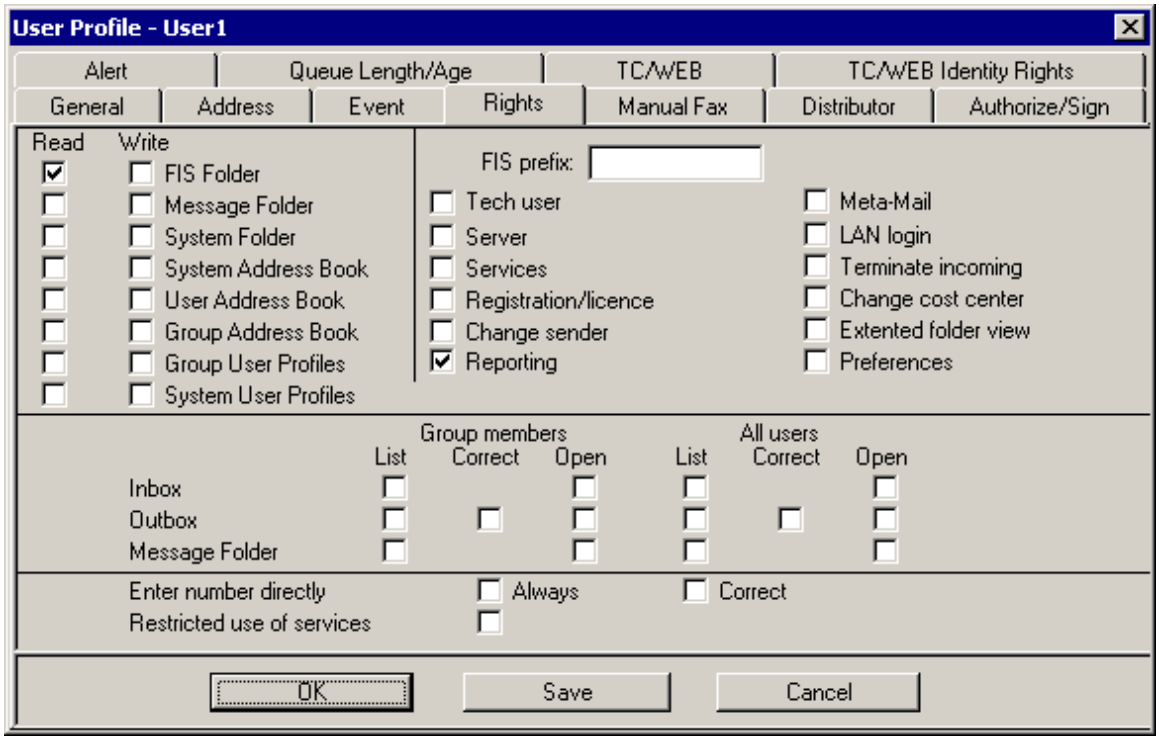


- **List Inbox** and **Outbox** of **All Users**
- **Optional**: **Open Inbox** and **Outbox** of **All Users profiles**:
  Only if **log entries**, **send attempts** or **message subjects** shall be fetched from this server

For TC/MA, the Fetch Agent's KCS user needs the following additional permissions:

- **Tech User**
- **Read System Address Book**
- **Read System User Profiles**
- **Read System Folder**
- **Open Inbox and Outbox of all users**

Permissions for users who request a report:



- For requesting a report a user must have the **Reporting** right.
- **Read FIS folder** right is needed the report request agent is used.

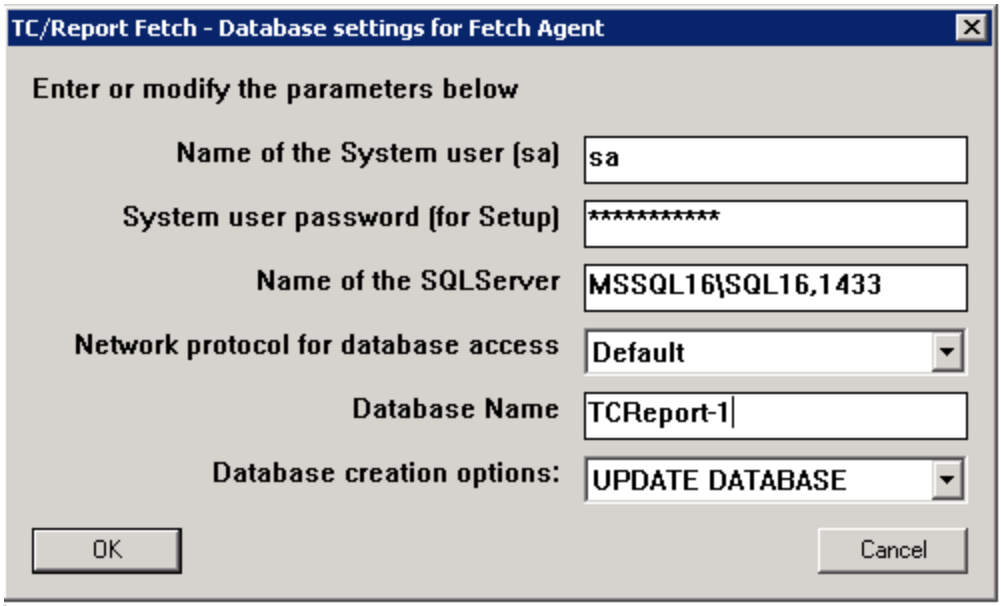
Step 3 – Setup Agents and Database

1. Make sure that the database is running.
2. Start KCS Setup. Fetch and Report agent can be installed separately.
3. Select TC/Report Fetch to install the Fetch agent and create the database user and tables.
4. Select TC/Report Report to install the Report agent, the Crystal Reports runtime and all standard reports.

Fetch Agent Setup

- Copies all necessary files for the Fetch agent
- Enters useful default registry settings (can be changed later via TC Management Console)
- Registers COM objects.
- Optionally starts a program to create the database, tables, and the SQL user that owns the database.

Screen 1 (Database Settings):



Name of the System user (sa)

If the TC/Report database does not yet exist, Setup needs system administrator permissions to create it. In this case, you must specify an SQL user with sysadmin permissions, such as, the built-in sa user. This information is needed for creating the user account for TC/Report and the TC/Report database.

If the TC/Report user and database exist already – and if any optional backup databases exist already - you can specify the TC/Report user instead.

System user password (for Setup)

Enter the password of this user. Default: no password.

Name of the SQL server

Enter the network name of the SQL server.

Note if SQL server is installed as a named instance, you have to specify the instance name as well. Syntax: <computer name>\<instance name>, see screenshot in section Report Agent Setup for an example.

For TCP/IP connections, you can add the port number after the instance name, separated by a comma.

Example: “LOCALHOST\SQLExpress,1254”

For connections to a remote MS SQL Server instance, it is necessary to specify the port number. The port number used by MS SQL Server can be found in the SQL Server Configuration.

To support SQL High Availability Cluster, use the following registry keys at `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\TOPCALL\TCPROBE`:

- MultiSubnetFailOver

- DBProvider
- ConnectionTimeOut

For more information about registry keys, see Fetch Agent Registry

Network protocol for database access

KCS Setup offers an option to choose the network protocol used for accessing the SQL server. The following protocols are possible:

- Default
- Win32 Named Pipes
- Win32 Winsock TCP/IP

Default: the default client access protocol configured at the local workstation. You can use the SQL Server Client Network Utility (cliconfg.exe) to configure the default protocol.

The chosen protocol must also be supported by the SQL server. The protocols enabled on the server can be viewed and changed via the SQL Server Network Utility (srvnetcn.exe) right on the SQL server.

The protocol you selected during KCS Setup is stored in REG_SZ registry value *General \DBNetworkLibrary* below the (Fetch or Report) Agent registry key.

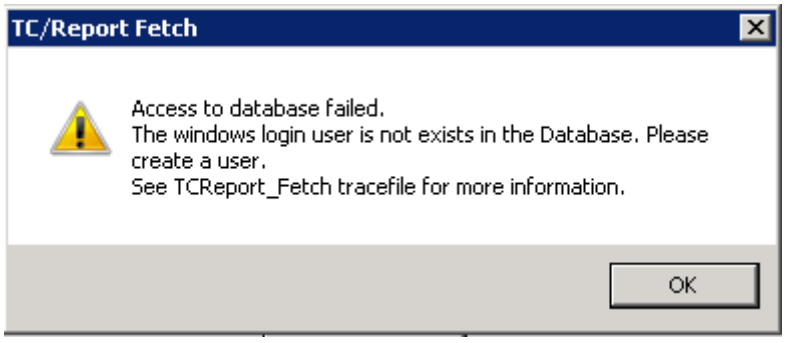
Possible values:

Value	Meaning
	Default
dbnmpntw	Win32 Named Pipes
dbmssocn	Win32 Winsock TCP/IP

Database name

Enter a name for the TC/Report database.

Note If the database name exists, specify the same name database name that you have created. If you use a different name, following error message is displayed.



Database creation options

TC/Report always needs a database structure that is up-to-date. Therefore, you must choose U**PDATE DATABASE for the first installation and for every version update**. Setup then updates all database items (user, database, tables, and stored procedures) according to the needs of the current TC/Report version.

If you want to remove the existing database including all tables and data and create a new one, choose **RECREATE DATABASE** (this works only if you do not change the database name).

If the current version of TC/Report is already installed, and want to reinstall TC/Report, you can choose **NO DATABASE CHANGE**.

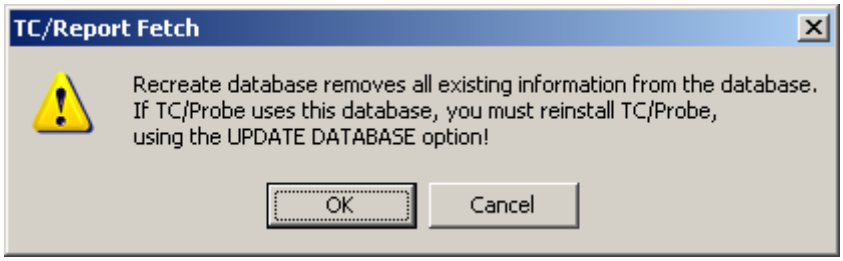
Attention:

It is possible that TC/Report and TC/Probe share the same database.

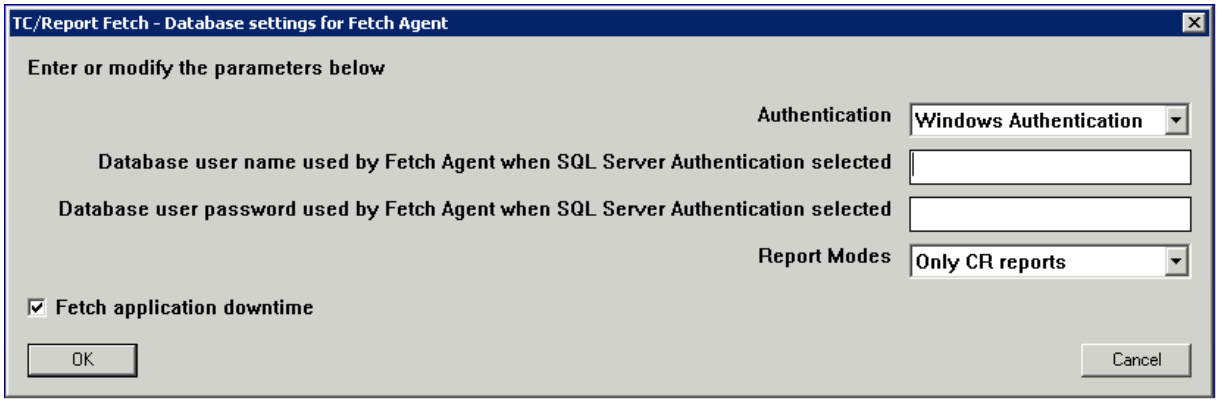
Recreating the database during TC/Report Setup removes the complete database. This deletes all entries in the database. Then, Setup reinstalls the database with all objects that TC/Report needs. Objects needed by TC/Probe only are NOT installed.

Therefore, you must reinstall TC/Probe afterwards, using the Update Database option during TC/Probe Setup.

Setup displays a warning message box before removing the database. Thus, you have a chance to cancel the action:



Screen 2 (Database Settings - Authentication):



Authentication

Select the authentication mode for accessing database: Windows Authentication or SQL Server Authentication.

Database user name used by Fetch agent

If you have selected the SQL Server Authentication, and creating database for the first time, you can specify a database user name. If the database already exists, do not change the user name.

Note This field is ignored if you select Windows Authentication for accessing the database.

Database user password used by Fetch agent

If you have selected the SQL Server Authentication, also specify the password for this database user.

Note This field is ignored if you select Windows Authentication for accessing the database.

Fetch application downtime

If selected, TC/Report can be used to monitor up and down times of KCS related applications. Setup will create additional database tables to hold this information, and the Fetch agent will evaluate the event log entries of these applications once a day.

Report modes

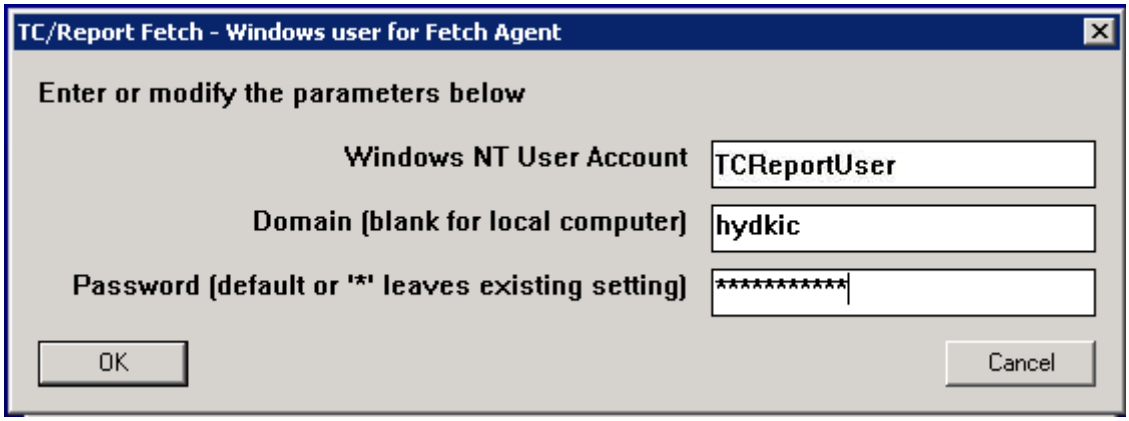
Setup now asks in which mode TC/Report shall be used:

- Only with OmniAnalyser (TC/MA license needed). In this mode, the tables Action_Table and Log_Table are not filled! This means no CR reports are available.
- Only with CR reports (Crystal Reports based or custom reports, TC/Report license needed)
- With OmniAnalyser and CR reports (TC/Report license needed).

For backward compatibility, the TC/Report license can replace the TC/MA license.

If OmniAnalyser integration is selected, Setup asks for additional configuration values concerning message tracking events.

Screen 3 (Database Settings – Windows Authentication):



If you have selected **Windows Authentication** in the previous screen, specify the following Windows credentials.

Windows NT user Account

Enter the name of the Windows user.

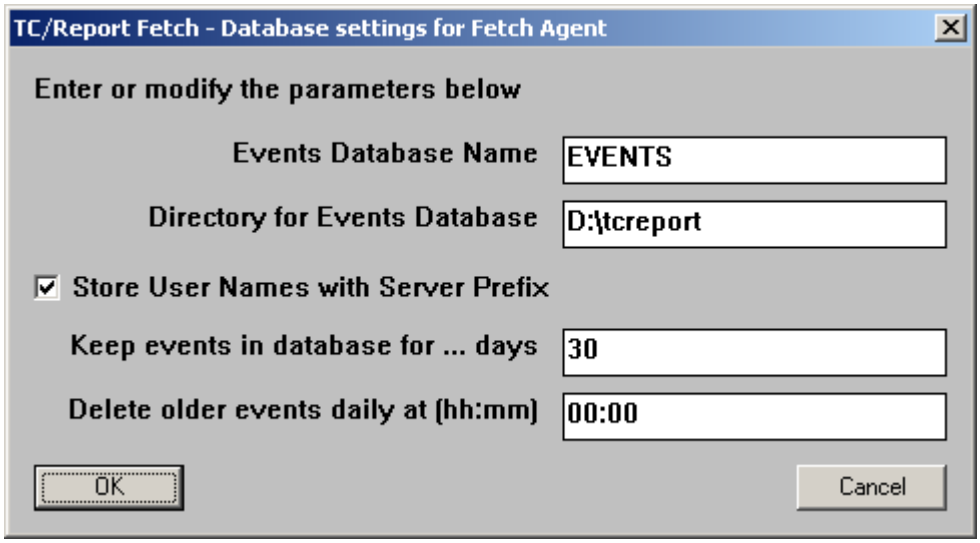
Domain

Enter the domain of the Windows user.

Password

Enter the password of the user.

Screen 4 (Database settings for Fetch Agent – Events database):



Events Database Name:

It is recommended to use a separate database for message tracking events, - as a different cleanup mechanism is used for the event tables and for the standard TC/Report tables: The oldest entries of standard TC/Report tables are deleted when the database size exceeds a configured value. The oldest entries of the event tables are deleted when they are older than a configured amount of days.

It is nevertheless possible to use the standard TC/Report database for events also. In this case, the second parameter (Directory for Events Database) is ignored.

Directory for Events Database:

If using a separate database for message tracking events, enter here the folder where the database shall be located. Default value: same as for TC/Report database.

Store User Names with Server Prefix:

You must select this checkbox if multiple TCOSS instances will be handled by one OmniAnalyser database. The effect is that user names in message tracking and directory will have the Server_ID (KCS short name) from Server_Table as a prefix, e.g. “DEMOTC.FS” instead of “FS”. Default: no

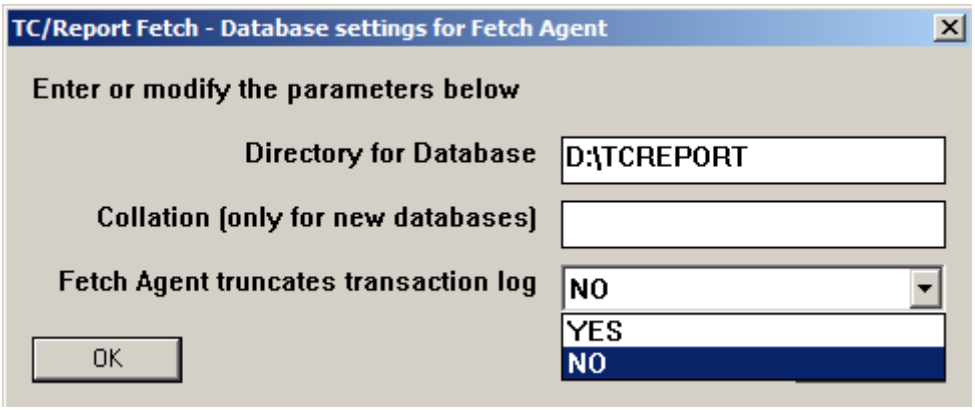
Keep events in database for … days:

Specify how long events shall be kept in the database until they are deleted. Default: 30 days

Delete older events daily at (hh:mm):

The Fetch Agent will perform event database cleanup daily at the specified local time.

Screen 3 (Database settings for Fetch Agent):



Directory for Database:

Enter the complete path name of the directory where the TC/Report database shall be stored. Setup expects you to enter the name of an existing directory.

Collation:

Setup now asks for the Collation name. If you want to use the default collation (matching the SQL server language), leave the text box empty. Otherwise, enter the correct collation name.

See Using TC/Report with Non-Standard Languages and Character Sets for details and possible values.

The collation name entered during Setup is stored in registry value *General\DBCollation* (REG_SZ).

Fetch Agent truncates transaction log:

Setup checks the SQL server version and offers this option only for versions up to SQL Server 2005. With SQL Server 2008 and above, this setting is always set to “NO”.

Possible values:
YES (General\ShrinkLog = 1), default, same as previous versions
NO (General\ShrinkLog = 0)

The registry value General\ShrinkLog is only used by Setup (and by TC/Report when creating a backup database). Changing the value without a subsequent “update database” Setup is useless.

If you use the second option (NO), the SQL Server administrator has to guarantee regular shrinking of the database log file, - e.g. by backing up the database via SQL Server Backup. If no regular backup is planned, the administrator can also manually change the recovery model of the TC/Report database to “Simple”, to avoid filling up the transaction log during bulk actions (channel definition changes, deletion of old records).

The database recovery model can be configured in the Options panel of the database properties (SQL Server Management Studio), or via the ALTER DATABASE command, e.g.:
ALTER DATABASE databasename SET RECOVERY SIMPLE

Database changes:

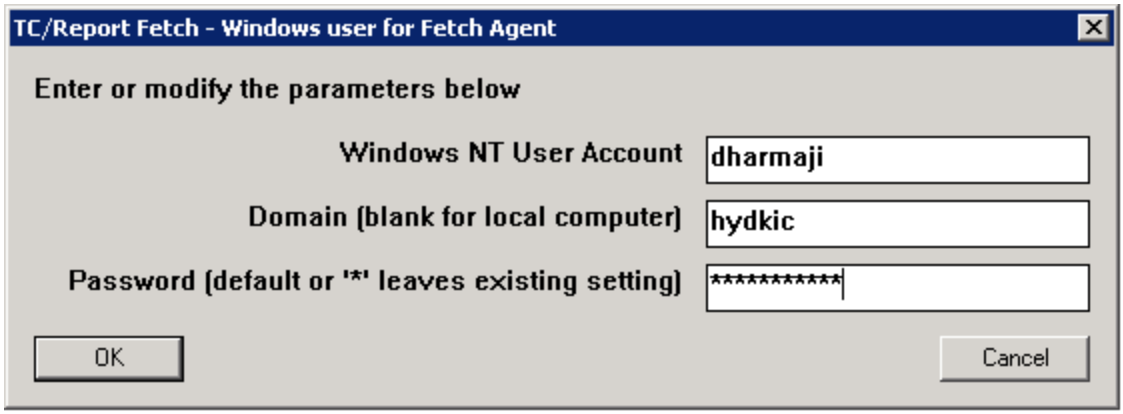
In the first releases of TC/Report, Setup started a script that upgraded the database. Now, this is done by a small program (TCREPSETUP.EXE), because a script does not offer the flexibility to have variable database locations, user names, and columns. Additionally, using a dedicated program allows installation of database objects on a remote SQL server.

If the database holds no tables yet (i.e. if the Server_Table is not found) or if the database shall be recreated, Setup installs a Unicode-enabled version of the Reporting tables, with most string columns defined as nvarchar (using 2 bytes for each character).

Simple database updates do not change the definition of string columns.

Note To enable Unicode support in an existing database, use TCREPSETUP in interactive mode as described in Database Upgrade to Unicode.

Screen 4: Windows user for Fetch Agent



This is the Windows user account for the TC/Report Fetch agent. This user account must be member of the **local Administrators** group and must have the right to **Log on as a batch job**.

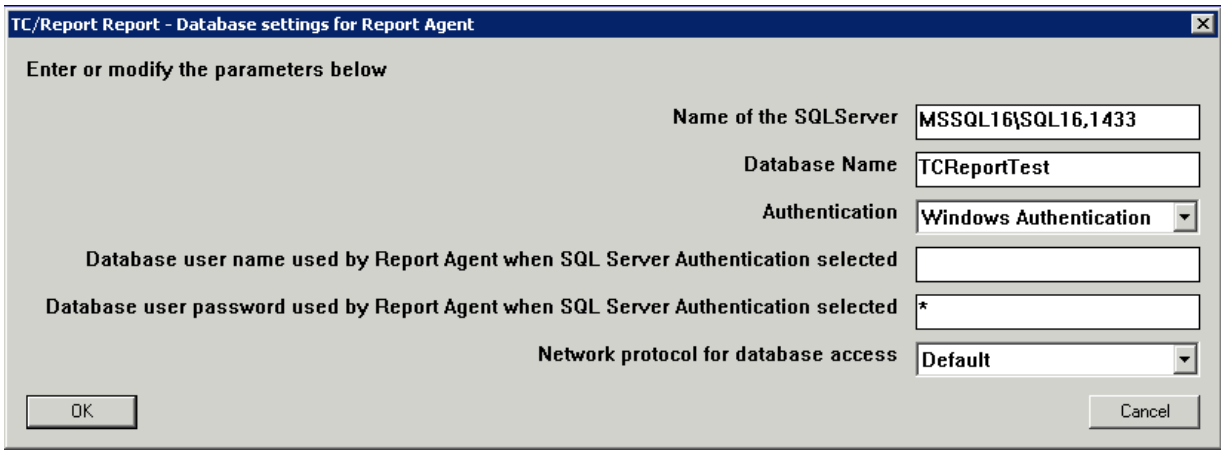
Note If you leave the account name empty, the Fetch agent will use the local system account.

Report Agent Setup

- copies all necessary files for the Report agent, including standard reports
- installs the Crystal Reports runtime (version 11.5)
- enters useful default registry settings (can be changed later via TC Management Console)

- registers COM objects.

Screen 1: Database Settings



Name of the SQL Server:

Enter the name of the computer hosting the SQL server. If SQL server is installed as a named instance, you have to specify the instance name as well.

Syntax: <computer name>\<instance name>,port

For example: PCFS\CASESENS,1433

To support SQL High Availability Cluster, add the following registry keys at `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\TOPCALL\TCPROBE` and set the described values:

- MultiSubnetFailOver: 1
- DBProvider: MSOLEDBSQL
- ConnectionTimeOut: 90

See Report Agent Registry

Database Name:

Enter the name of the database used by the Fetch Agent.

Database user name used by Report agent:

This must be the user name specified when the database was created via the Fetch agent setup.

Note This field is ignored if you select **Windows Authentication** for accessing the database.

Database user password used by Report agent:

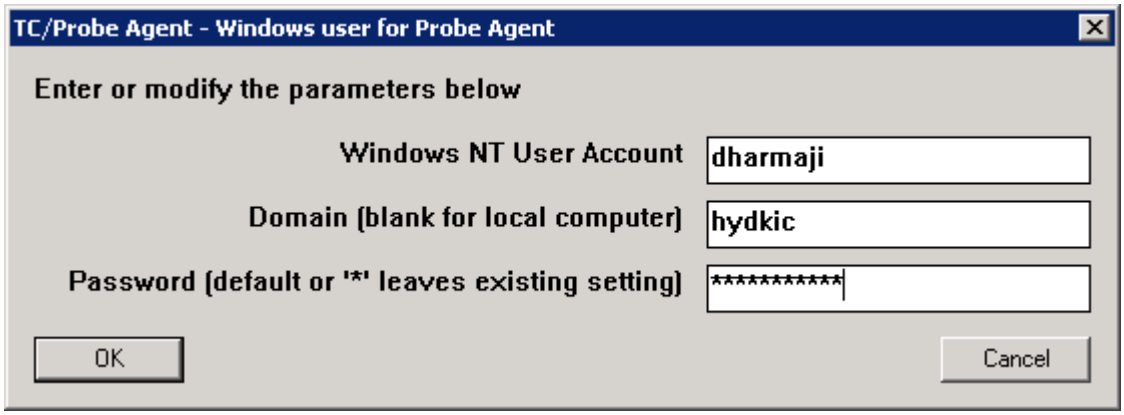
Enter the password for this database user.

Network protocol for database access:

Here you can choose a network protocol. See Fetch Agent setup for details.

Note Report Agent setup does not create any objects on the SQL server. Therefore, you should first install the Fetch Agent (on any computer). Afterwards, install the Report Agent and specify the database name, user name and user password used during Fetch Agent installation. (If both agents are installed on the same computer, Setup offers you the settings configured for the Fetch Agent as a default).

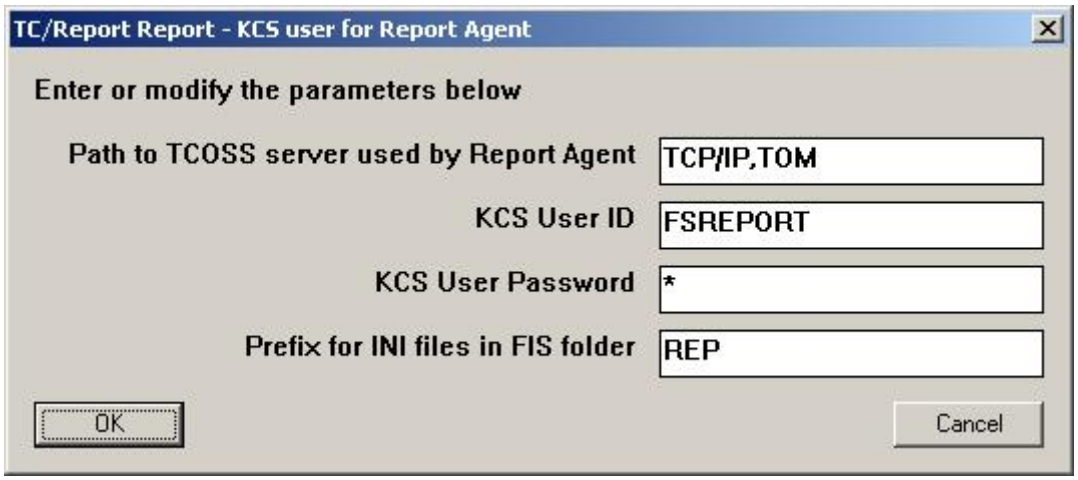
Screen 2: Windows user for Report Agent



This is the Windows user account for the TC/Report Report agent.

It must be member of the local **Administrators** group and must have the right to **Log on as a batch job**. Additionally, this account should have **access** to **all network drives and printers** that will be used as report destinations.

Screen 3: KCS user for Report Agent



Path to TCOSS server used by Report Agent:

Enter the path of the TCOSS server where report requests shall be retrieved.

KCS User ID:

This is the name of the TC/Report KCS user profile. This user must be configured as a gateway user (visible in outbox flag set) and must have read access to the System Address Book, System and Group User Profiles and (optionally) read and write access to the FIS folder.

KCS User Password:

Enter the password this user.

Prefix for INI files in FIS folder:

The Report Agent will copy the report INI files to the FIS folder. To avoid deletion of existing FIS documents, choose a unique prefix for them. If you leave this field empty, the Report Agents will not copy the INI files to the FIS folder.

Messagebox concerning CRPE32.DLL

Setup may display a warning that CRPE32.DLL already exists on the workstation. If this is an upgrade of an existing and running TC/Report installation, just overwrite the existing file.

If this is a new installation, back up a copy of the existing file and then let Setup install the new file version.

Background: Some application seems to use a library called CRPE32.DLL that is totally incompatible with the Crystal Reports CRPE32.DLL used by TC/Report.

Step 4 – Configuration

Start TC/Management Console locally (TCOSS login is not necessary). Please note that currently remote configuration is not possible.

Usually the TC/Management Console connects to the database. Make sure that the Fetch Agent and the Report Agent are not running while TC/Management Console is connected to the database.

After the first Setup, you must at least configure which TCOSS servers shall be fetched.

See section Configuration via TC Management Console for details.

Cost Accounting

Fax and Telex costs are configured on TCOSS. Please refer to the TCOSS manual for details.

Configuring Costs for the TC/LINKs: Make sure to enable Cost accounting (*General\CostsPerRecipient*) only for that TC/LINK's, for which you did an assignment (Link queues) to a specific message class. Otherwise these costs are reported in a Message_class column without description (unassigned). Doublecheck if the CostsPerRecipient is set to 0 for all other TC/LINK's for which you want to do no cost accounting.
For details please refer to the TC/LINK manual.

License for TC/Report

The TC/Report Fetch agent requires a license. The license can be entered via licences.exe (installed with KCS setup). The Report agent does not need a separate license.

You need one license for each workstation where TC/Report Fetch agent is running (usually one) on each TCOSS server fetched by TC/Report.

For OmniAnalyser integration (TC/MA), a special license for KCS Messaging Analyser is available. This license is counted per workstation. If you use only OmniAnalyser for reporting, this license replaces the standard TC/Report Fetch Agent license. Please note that several tables needed for standard Crystal Reports based reports are NOT filled in this mode (e.g. Action_Table, Log_Table).

Final Installation Steps

1. Check necessary page file size
2. Make sure that the process user has the Windows User right **log on as a batch job.**
3. Switch TCSRV service to automatic startup.

Configuration for KCS Tandem System

You can configure TC/Report to connect to a KCS Tandem System. In this case, the KCS path for both agents must contain the path of the primary master and the path of the secondary master, separated by a vertical bar '|' (the *or* sign).

Example: TCP/IP,PRIMARY_SERVER|TCP/IP,SECONDARY_SERVER

If the primary master is unavailable, TC/Report is automatically connected to the secondary master.

In the case of a (very unlikely) desync condition of a KCS Tandem server logging entries can get lost.

Configuration for Country Codes (rr99)

For correct recognition of country codes the normalization of fax numbers must be configured correctly.

If you want to see e.g. "43" for outgoing messages as cc_out, you have to configure the rr99 e.g.:

```
**NORMALIZE
F:*~,F:*~,                already normalized
F:00~,F:*~,               international
F:0~,F:*43~,              Austria
F:I~,F:*43186353~,        internal
F:~,F:*431~,              Vienna
**ROUTE
F:*43186353~,F:<I>~
F:*431~,F:~
F:*43~,F:0~
```

```
F:*~,F:00~
```

Additional Configuration for Application Monitoring

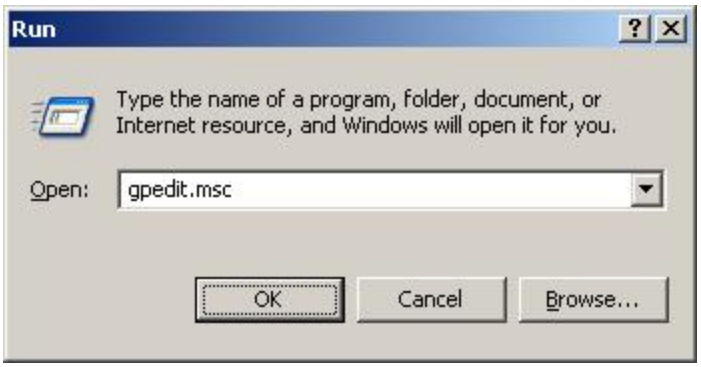
If TC/Report is used for application downtime monitoring, it is necessary to install a shutdown script on all monitored workstations. This script guarantees that event 21534 (Shutdown of the service control manager) is written to the application event log when the system is shut down.

KCS Setup copies the following 2 files to directory C:\TOPCALL\SHARED:

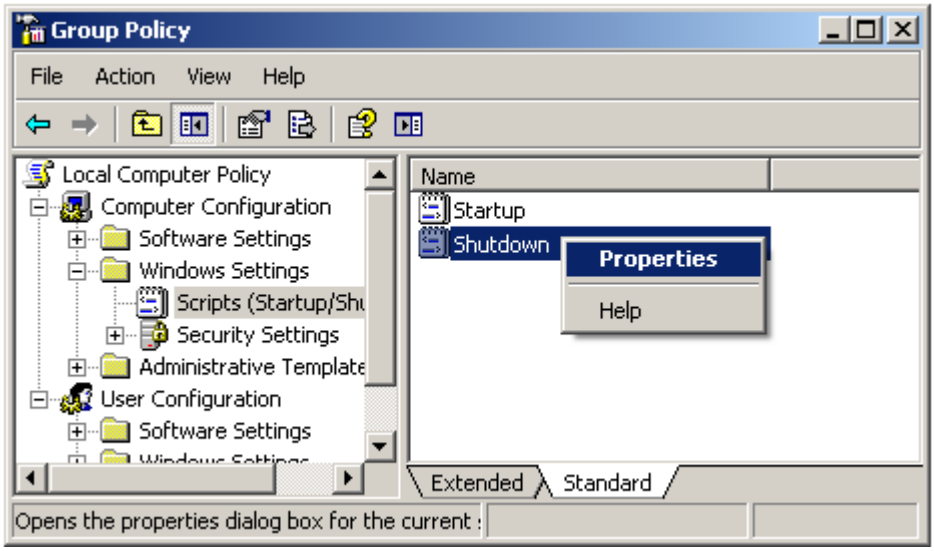
LogWinEvent.exe	(the application that writes event 21534)
LogStop.bat	(a simple batch job that invokes LogWinEvent.exe)

On every computer that shall be monitored by TC/Report application downtime monitoring, install these files as follows:

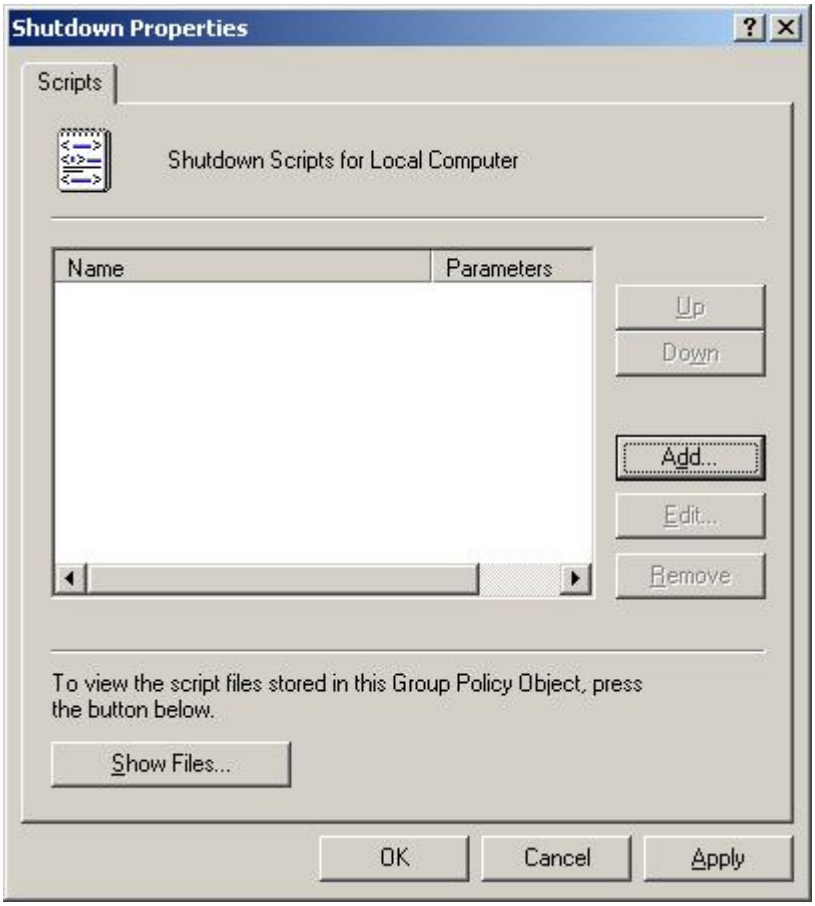
1. Make sure that LogStop.bat and LogWinEvent.exe are in the C:\Topcall\Shared folder, or in another folder that is within the path for all users.
2. Open the Group Policy Editor (gpedit.msc):



3. Open the Scripts container below **Computer Configuration | Windows Settings**. Select the Shutdown script and select Properties from the context menu:

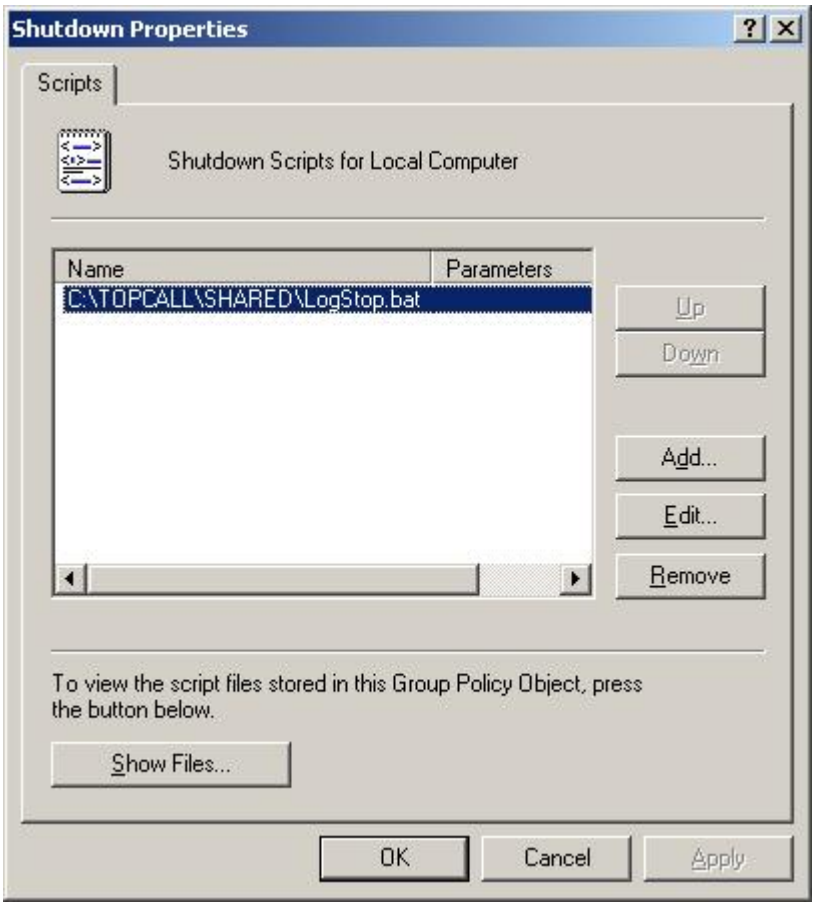


4. Click **Add** in **Shutdown Properties**:



5. Browse for the LogStop.bat file, or enter its full path name in the **Script Name** field:

6. Click **OK** and close the group policy editor:

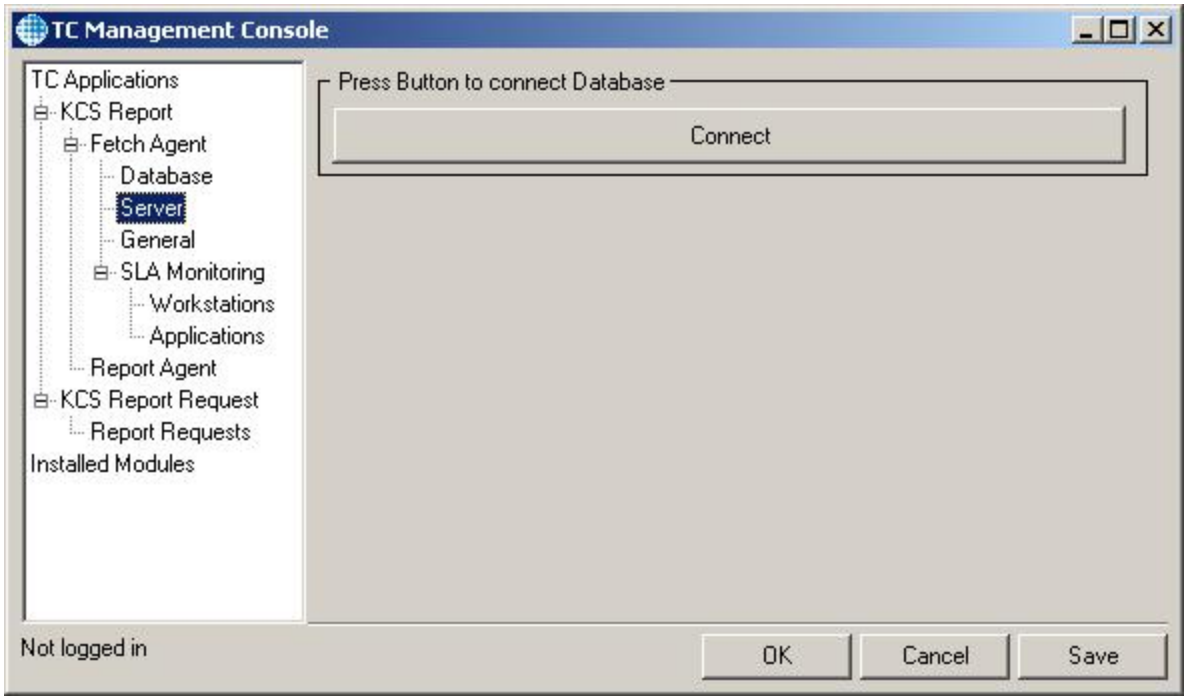


Chapter 5

Configuration via TC Management Console

Connecting to the database

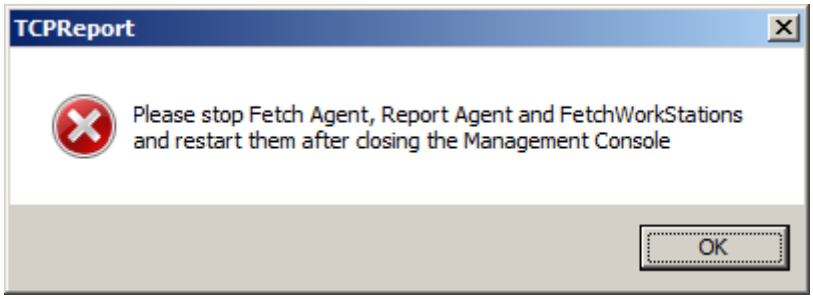
After starting the TC Management Console, the property pages for **Servers**, **Workstations** and **Applications** show a **Connect** button. As soon as one of these **Connect** buttons is clicked, the tool connects to the database and populates all three property pages with data.



Saving changes to the database

Please note that any changes are only saved when clicking either the OK button or the Save button at the bottom of the TCMC user interface. (In versions before KCS 9.2, some changes were effective immediately.)

The TCREPORT agents must be stopped while changes are saved to the database. Before writing data to the database, TCMC brings up a message box that reminds you to stop these applications.



Windows Server 2008 or Later

When running on a Windows Server 2008 or later operating system, TC Management Console must be started with option “Run as administrator”. Otherwise, configuration changes are stored in the logged-in user’s registry section and do not affect the TC/Report agents.

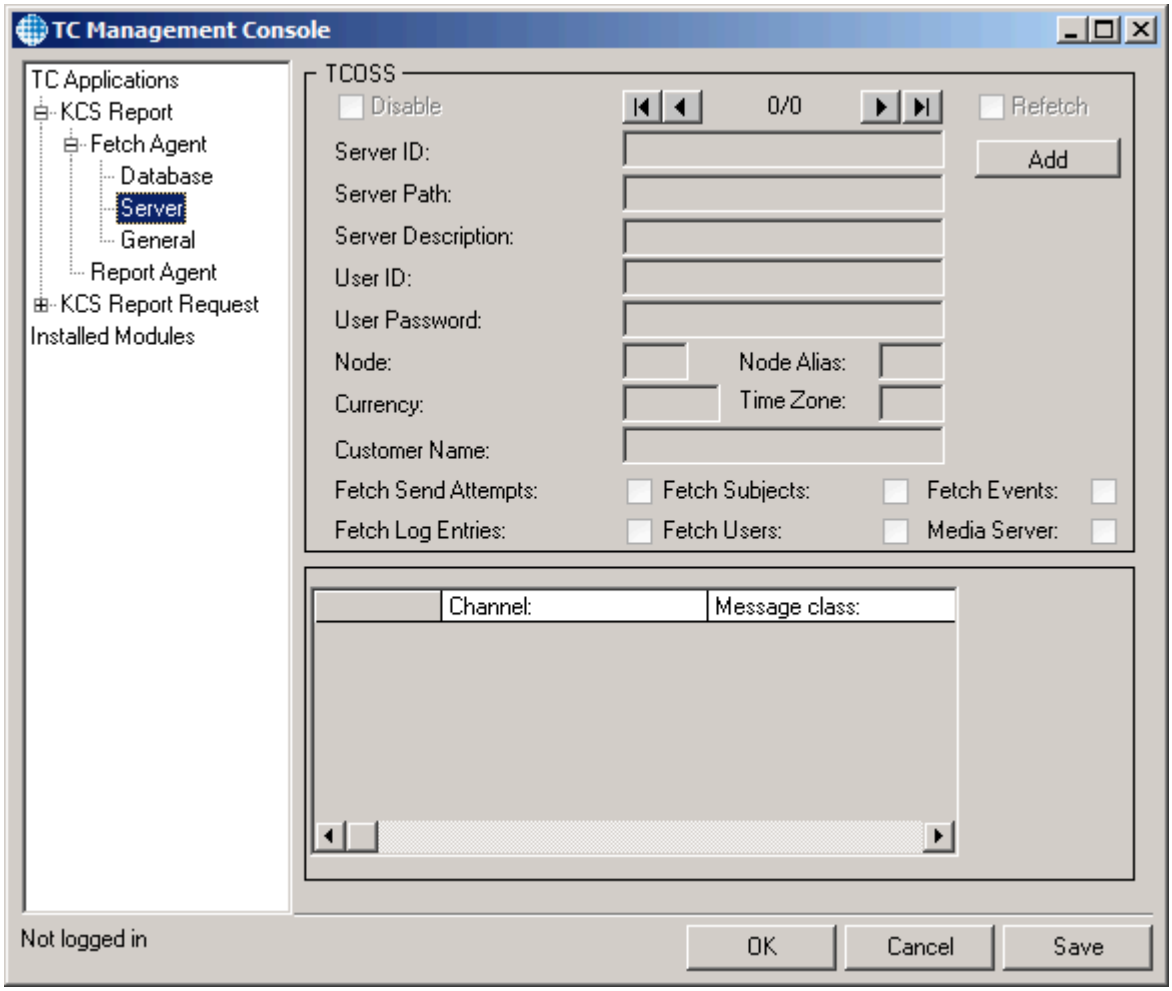
Fetch Agent Configuration

The Fetch Agent container in the TC Management Console holds the several property pages.

Servers Property Page

The Fetch Agent can fetch messages from several TCOSS instances at the same time.
After first Setup, you must use this property page to tell TCReport which TCOSS instances shall be fetched.

If no servers have been configured yet, the property page looks like this:

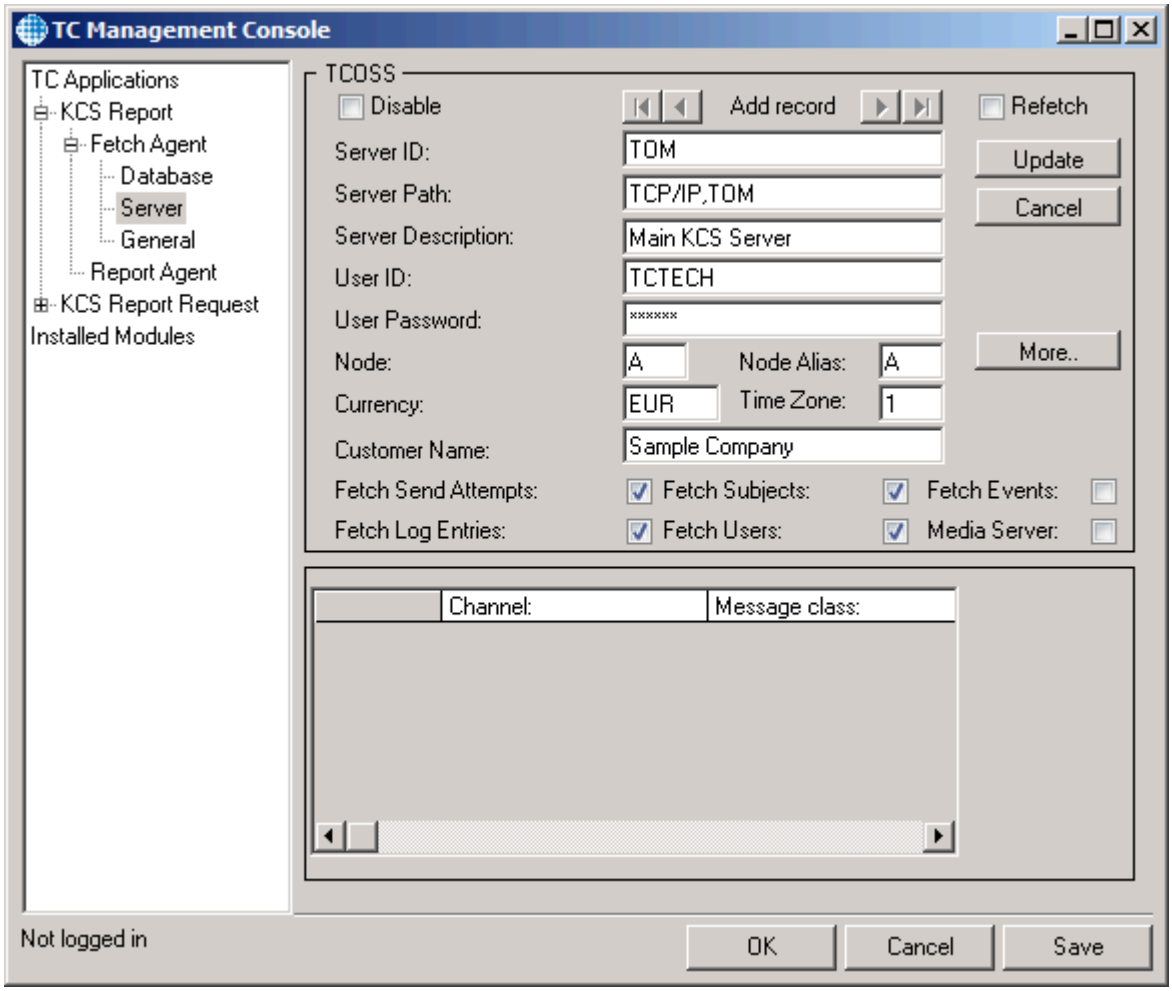


Add

To add a new server definition, click the **Add** button in the upper part of the screen. The new server may be a TCOSS instance you want to collect data from, or a media server.

After clicking **Add**, you are in Edit mode and can define the characteristics of a TCOSS instance to be used by TC/Report.

General server parameters



Server ID: Any desired name, but preferably keep it short.

If TCSNMP is installed on the Fetch Agent computer, it is recommended to use ASCII strings. TCSNMP encodes characters that are not part of the local Windows code page as hex digits, thus making the names unreadable. This encoding makes the string longer. For compliance with the SNMP protocol, TCSNMP truncates encoded strings that are longer than 100 bytes. This means that pure Unicode names can be truncated to 12 characters. More information about encoding can be found in the TCSNMP manual, section “TC/SNMP Advanced Settings”.

Server Path: can be an alternative path, for example: TCP/IP,PRIMARY|TCP/IP,SECONDARY

Server Description: This name is displayed under Server Name at the top of a returned report. Keep this name short as well to keep it from being truncated, especially when you’re requesting a report of various servers, since the designated field in the report is not too long either.

User ID: A KCS user account that will be used to connect to the TCOSS server.
The permissions needed for this account are described in section 4.2 (Permissions for the Fetch Agent).

User Password: of user to be connected to the TCOSS server. The password is stored encrypted.

Node: (node character) Default “A”; other values may be used only if LCR via TCP/IP is configured on TCOSS.

Time Zone: of the TCOSS server (in hours). The Fetch Agent calculates the UTC-time by subtracting the time zone parameter from the local time. Possible values are –12 to 12. **Example**: for Central Europe Time zone the value is 1.This setting is not used in the standard reports.

Currency: The currency is necessary if you want to request reports of TCOSS servers with different currencies. Only some reports use the currency.

Customer Name: In an ASP environment, you can use this field to correlate the TCOSS instance to a customer. The customer name can appear on the report.

Fetch Log Entries: TCOSS 7.41.00 and above stores log entries in the short term archive. TC/Report uses them for the report types QUEUELEN, TCJOB and TCJOBDL. The Fetch agent must be explicitly configured to fetch log entries.

Fetch Send Attempts: TCOSS 7.41.00 and above stores send attempts in the short term archive. Use this checkbox to configure fetching of send attempts.

Fetch Subjects: Fetch also the Subject of a message. You must restart the fetch agent to make changes effective. Note: If the Fetch Agent KCS user does not have the right to open all in- and outboxes, only the first 32 characters of the subjects are available.

Fetch Users: Fetch user information from this server.

Fetch Events: Fetch message tracking events from this server (for TC/MA only).

Media Server: Use this checkbox to indicate that this is a media server.
For media servers, you should disable all Fetch checkboxes, but use the **HW Channels** button described below to import the list of fax and telex channels that are available on this media server. The media server name and its channels can then be used as grouping criteria in reports.

Disable: if you disable a server, the fetch agent doesn’t fetch entries from this server. The user will still log in to it though. If this server should be temporarily down, this will result in login errors in the report agent’s trace file.

Refetch: If this option is checked, all fetched entries of this server will be deleted from the database and the Fetch Agent will start fetching with the oldest entry of the short term archive. While there are records to fetch, the configured poll cycle is ignored (no delay).

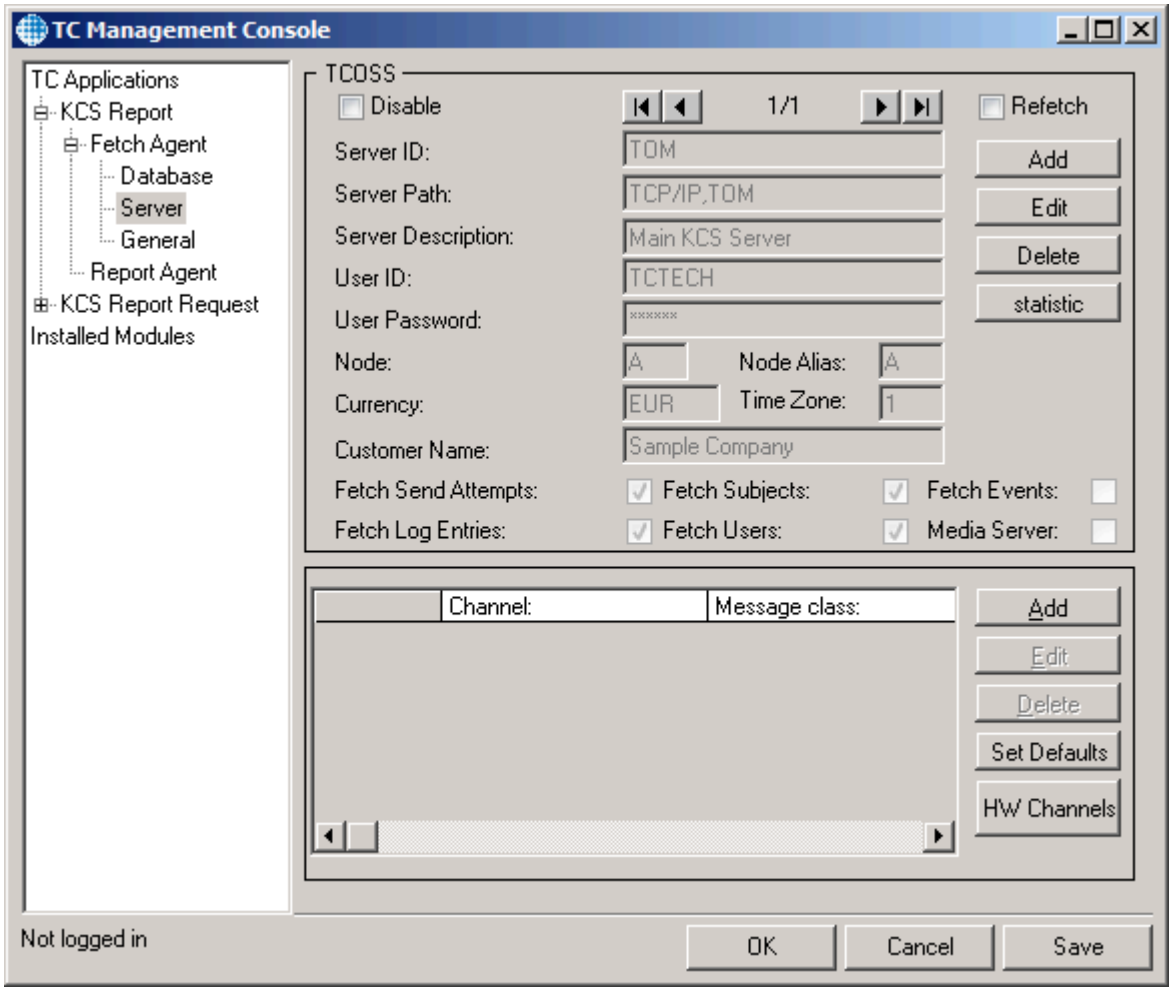
More: the More button leads to additional parameters that are needed for Probe Agent integration. See the Probe Agent manual for details.

Update: Use this button to finish configuration of the general parameters of this server.

Cancel: Use this button to cancel your changes.

Configuring channels and message classes

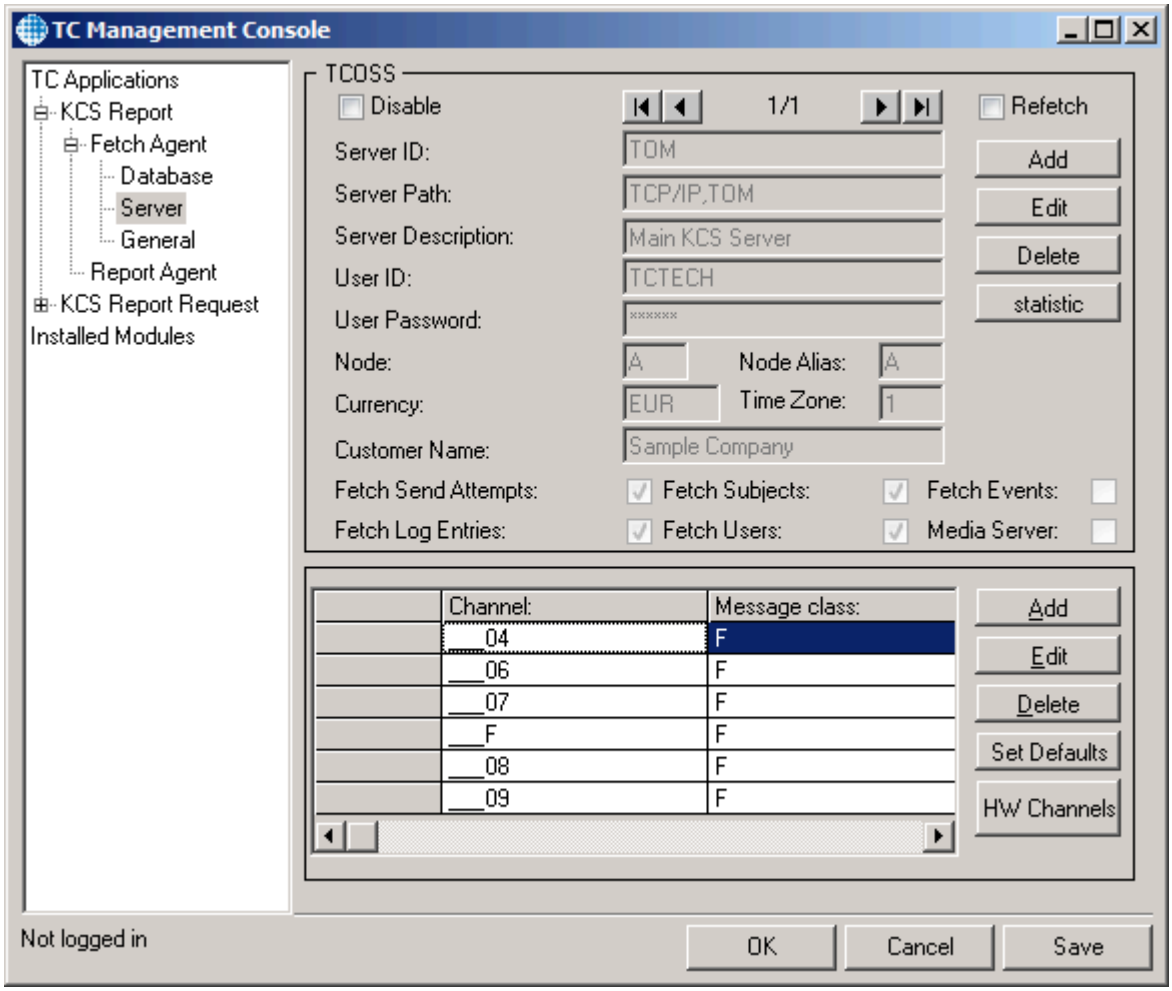
By clicking Update or Cancel, you leave the Edit mode. You now can configure the channels and message classes (services) available on this TCOSS instance.



HW Channels

With the **HW Channels** button, you can load fax and telex channel definitions from TCOSS. This should be done for normal TCOSS servers and also for media servers!
Information about fax and telex channels is needed for the report types LINES and TCLINESN, as well as for certain features of the UNIV report (grouping by country code or channel group).

A login screen comes up where you must enter the credentials of a user on this server. This user must have the right: "**Read System Folder**" and "**Tech User**". TC Management Console then reads the TAM config files on the TCOSS server and extracts the information about fax and telex channels and their channel groups.



The channel definitions created by TC Management Console consist of a channel name starting with "___" (three underscores) followed by the real name of the channel or channel group, and a message class that is either "F" for fax or "T" for telex.

These assignments are stored per server in the **Channel_Table**.

This automatic load of configuration does not lead to an update of the Action Table (this is not necessary for hardware channels).

Example for country code assignment

Fetch Agent processes a send order with normalized sender "F:*4316613351".
In the **Channel_Table**, there is an entry matching channel "___F" to message class "F". This tells the Fetch agent, that the send order was received via a physical channel and that it was a fax channel (message class "F").

Therefore, a lookup is done in the **Country_Codes** table, looking for a fax country code that matches the normalized sender number. The country code is supposed to be preceded by a '*'. So, the Fetch Agent searches for the best matching country code of class 'F' for the number "4316613351" and finds country code "43" (Austria). The resulting country code "43" is stored in field **cc_in** in the **Action_Table**.

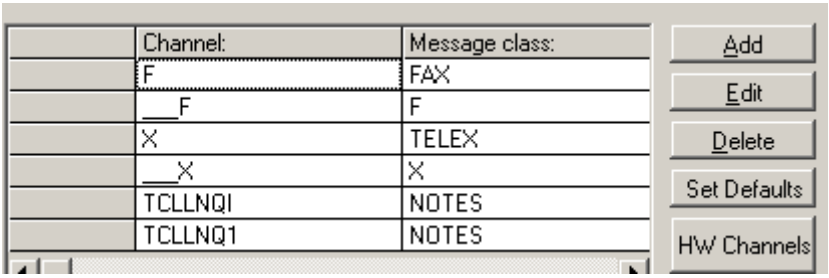
Message classes

Some reports (e.g. the UNIV report) can show message traffic grouped by message class, for example all messages sent to SAP.

The term message classes is used here more or less equivalent to KCS services. Use **Add** button to configure assignments between TCOSS channels and message classes. Use **Edit** and **Delete** to edit or delete existing message class assignments.

These assignments are stored per server in the **Channel_Table**.

Example:



Channel	Message class	Comment
F	FAX	Message class assignment for channel group
04	FAX	Message class assignment for single channel
05	FAX	Message class assignment for single channel
TCLSMQ*	SMTP	Message class assignment for link queues.

Matching channels and message classes:

For reports grouped by message classes (possible in UNIVERSAL and UNIV report), define message classes and assign channels and channel groups to these classes. It is allowed to enter wildcards ('*') in the channel, but it must be a unique name, e.g. the combination TCLN* and TCLNQ* is not allowed.

For TCOSS groups, e.g. Fax group "F", specify the group and all channels belonging to the group for the message class FAX in order to cover all messages, otherwise messages sent directly to a channel (e.g. to "05:1234" where 05 is a fax channel) are not part of the message class FAX costs.

In the above screen shot, all lines except those for channels starting with '___' are typical message class assignments. The channel definitions starting with '___' were created automatically in the previous step.

Note When you change manually the assignment Message class – TCOSS channel/group the Fetch Agent will update the message class information for already fetched messaging data in the TCReport database automatically (but not the backup databases). The former message class assignments for send orders from all servers (also disabled servers) are overwritten.

This update is started as soon as the Fetch Agent is aware of it. The update can take about 20 minutes (depends on the database size and speed) and is done without intermediate fetch delays. The KCS Monitor displays the state **Updating Database**.

Editing an existing server definition

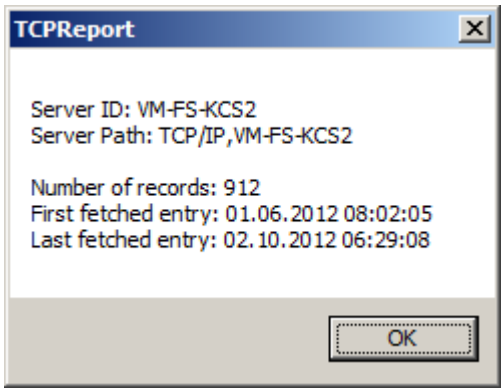
To edit an existing server definition, click the **Edit** button in the upper part of the screen.

Deleting an existing server definition

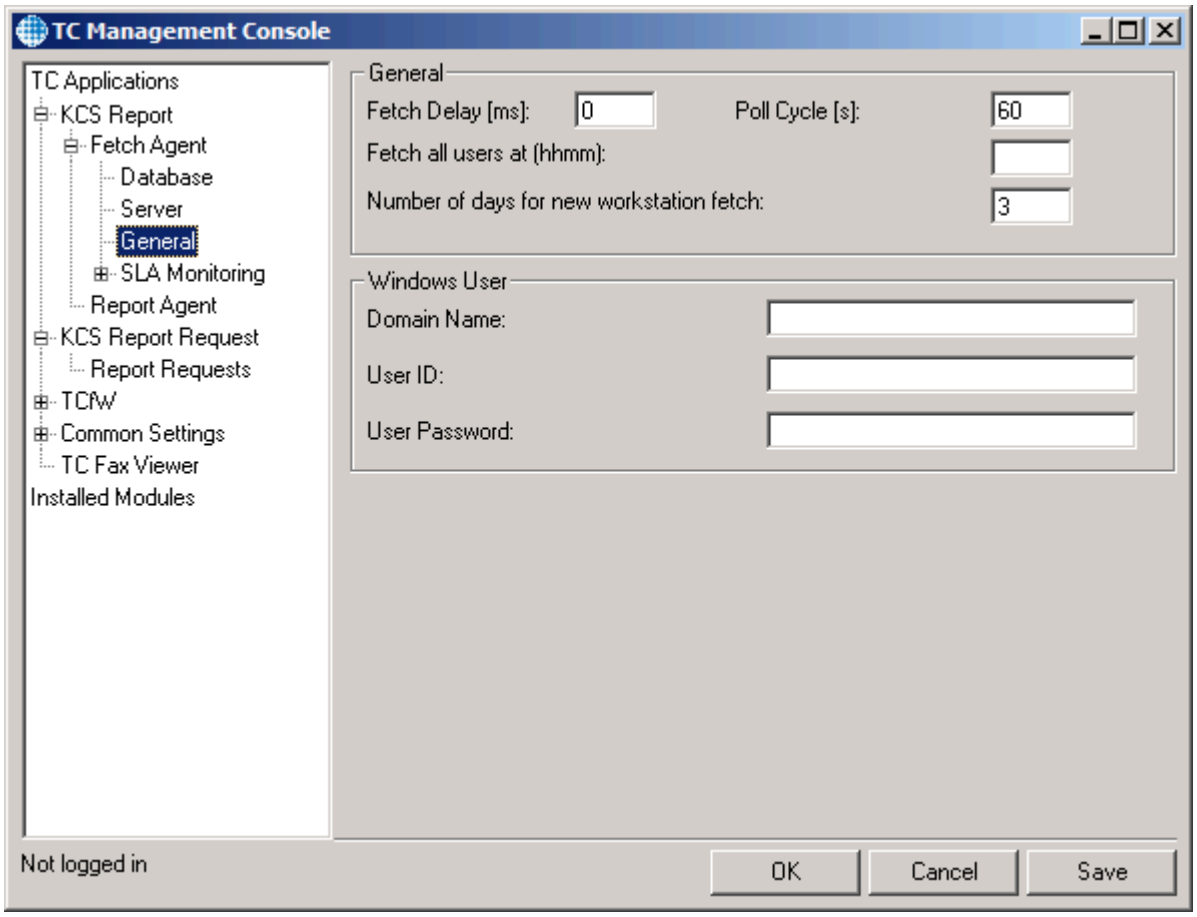
The **Delete** button in the upper part of this screen deletes the server from the database, and also all information about messages, log entries and users from this server. The information is not removed from backups that were previously created. Deletion takes place after clicking OK or Save.

Statistic

The button **Statistic** shows short information about message entries fetched from this server:



General Property Page



The following parameters apply to all fetched servers:

Fetch Delay: After each 100 fetched entries the fetch agent waits the amount of time configured in this field (milliseconds). During a refetch of a server, it is recommended to increase the value fetch delay to e.g. 1000 (milliseconds), because continuous fetching without a delay needs much performance of the SQL server.

Poll Cycle: Once the download of the short term archive of all servers is complete, the fetch agent polls the servers for new entries in the interval entered here. This means, that if the fetch agent is busy fetching all the data from the short term archive the value in this field will be ignored. You won't get a faster download if you reduce the poll cycle, because the agent can't retrieve data any quicker than it becomes available.

Fetch all users at (hhmm): If fetching of user profiles is enabled, you can use this field to configure a daily full fetch of all user data at a certain time. If this is not wanted, leave the field empty.

Note When the **Poll Cycle** or the **Fetch Delay** are changed while the Fetch Agent is running, it has to be restarted.

These parameters, **Poll Cycle** and **Fetch Delay**, are not used when the Fetch Agent is updating the database (after setup of TC/Report Release 1.01.09 or when the assignment channel, message class is changed).

Number of days for workstation fetch: This option is only available if you selected application downtime fetching during Setup. Here you configure the amount of information fetched from the event log of newly configured workstations (see below).

In the lower part of the screen, you see the user account configured for the TC/Report Fetch Agent. Normally, this account is configured during Setup and need not be changed afterwards.

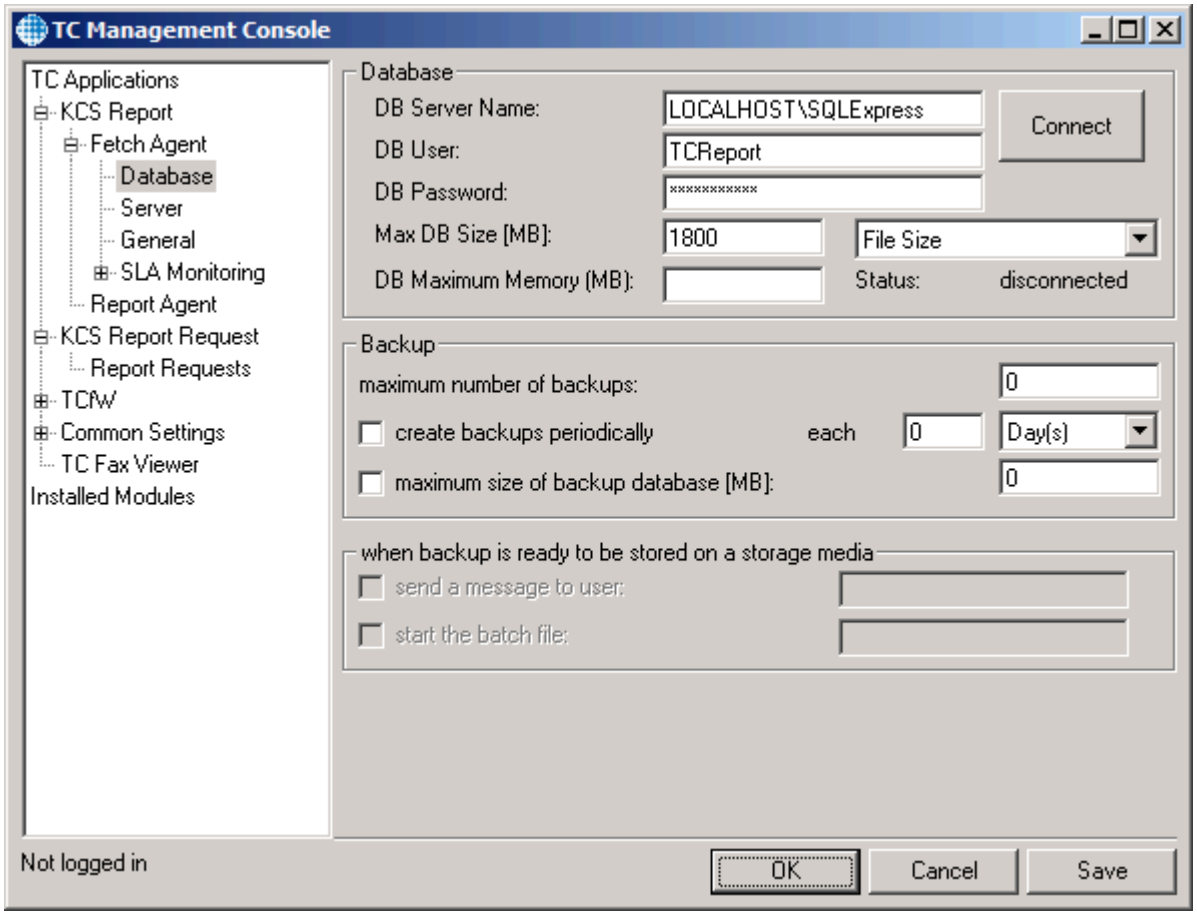
Domain name: Domain of this user

User ID: name of this user

Password: password of this user (displayed and stored encrypted)

Database Property Page

Here you can set parameters used for connecting to the database:



DB Server Name: computer name of the Database Server

DB User: user to be connected to database

DB Password: this user's password

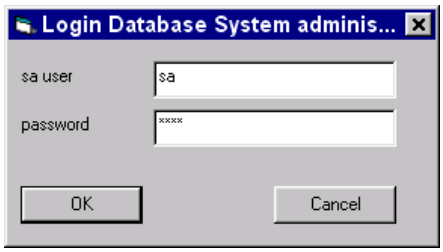
Max DB Size (MB): maximum size of the database. To disable database size check, set this value to 0. TCReport offers two ways of handling the database maximum size. By default, it checks the physical size of the database file, and stores the configured maximum value in registry value DBMaxSize.

For databases with fixed size, use the other available list box value, Content Size. This limits the amount of data stored in the database. The maximum content size is then stored in registry value DBMaxDataSize.

It is recommended to configure a value that is 10 % less than the available space. TC Management Console displays an error message if the value conflicts with individual file size limits.

DB Maximum Memory (MB): Maximum RAM usage of SQL-Server

To configure the maximum memory of the SQL-Server, you need a connection with the database system administrator account (“sa”). Click the Connect button and enter the password of the “sa” user.



In the lower part of this screen you can define a backup strategy for the TCREPORT database. Database backup is described in detail in section “Backup of the Database”.

Maximum number of backups: Select how many backup files shall be created (if needed, the oldest files are overwritten)

Create backups periodically: Here you can enable backup

Each: select the backup interval (a defined number of days, weeks, months, quarter years, half years or years).

Maximum size of backup database (MB): Define the maximum size in MB you want the backup database to reach before opening a new one (in case more than one is defined) or overwriting the existing one (if only one is defined).

when a backup is ready to be stored on a storage media (e.g. when the configured time period is fetched)

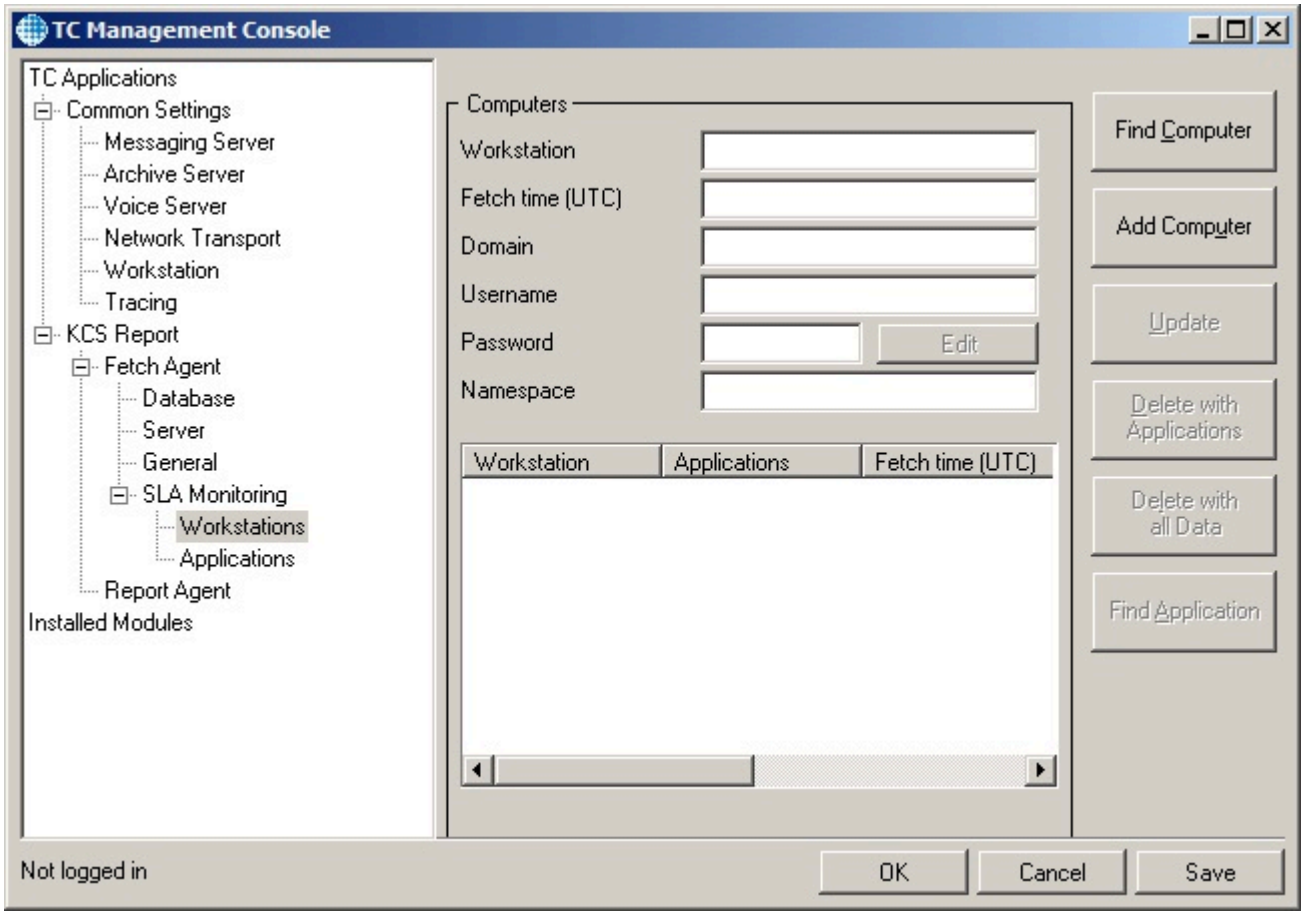
- the fetch agent can **send a message to a user** (enter service and username)
- the fetch agent can **start a batch file** (with the following parameters:
  Name of the backup, DateFrom, DateTo)

SLA Monitoring Property Pages

If Application Downtime Fetching was selected during Setup, the TC Management Console contains an additional item **SLA Monitoring** with sub-items **Workstations** and **Applications**.

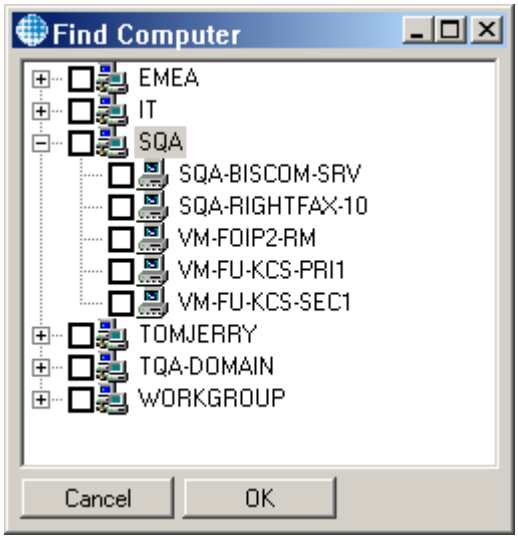
Please note that changes done in the SLA Monitoring Property Pages are only stored when clicking the **OK** button or the **Save** button.

The Workstations panel shows a list of workstations hosting monitored applications. The settings for the list item currently in focus are displayed above the list and can be edited there.



Adding monitored computers

Right after installation, this list is empty. You can fill it via **Find Computer** and **Add Computer**.



Find Computer opens a dialog that displays the domains and workgroups of the network neighborhood. You can select one or more computers by clicking the checkboxes beside the computer names. It is also possible to select all computers of a domain by clicking the checkbox beside the domain name.

Prerequisites:

“Find Computer” works only if the local “Computer Browser” service is running. If running on Windows Server 2008 or later, Network Discovery must also be enabled. On these operating systems, both services are by default disabled.

You must be logged on as a domain user who is member of the remote machine’s Administrators group, or you must connect to the remote computer explicitly, using one of the remote computer’s administrator accounts (via Start | Run | \\machine-name).

The machine name must be exactly the same as within "find computer" - no domain name or IP address is allowed.

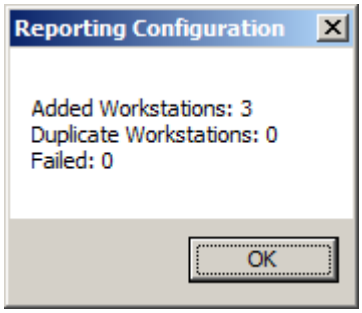
Error Handling:

If an error occurs while looking for domains and domain members, an Errors button appears on the bottom of the window. Clicking on the errors button opens a message box showing all errors that occurred, e.g.:

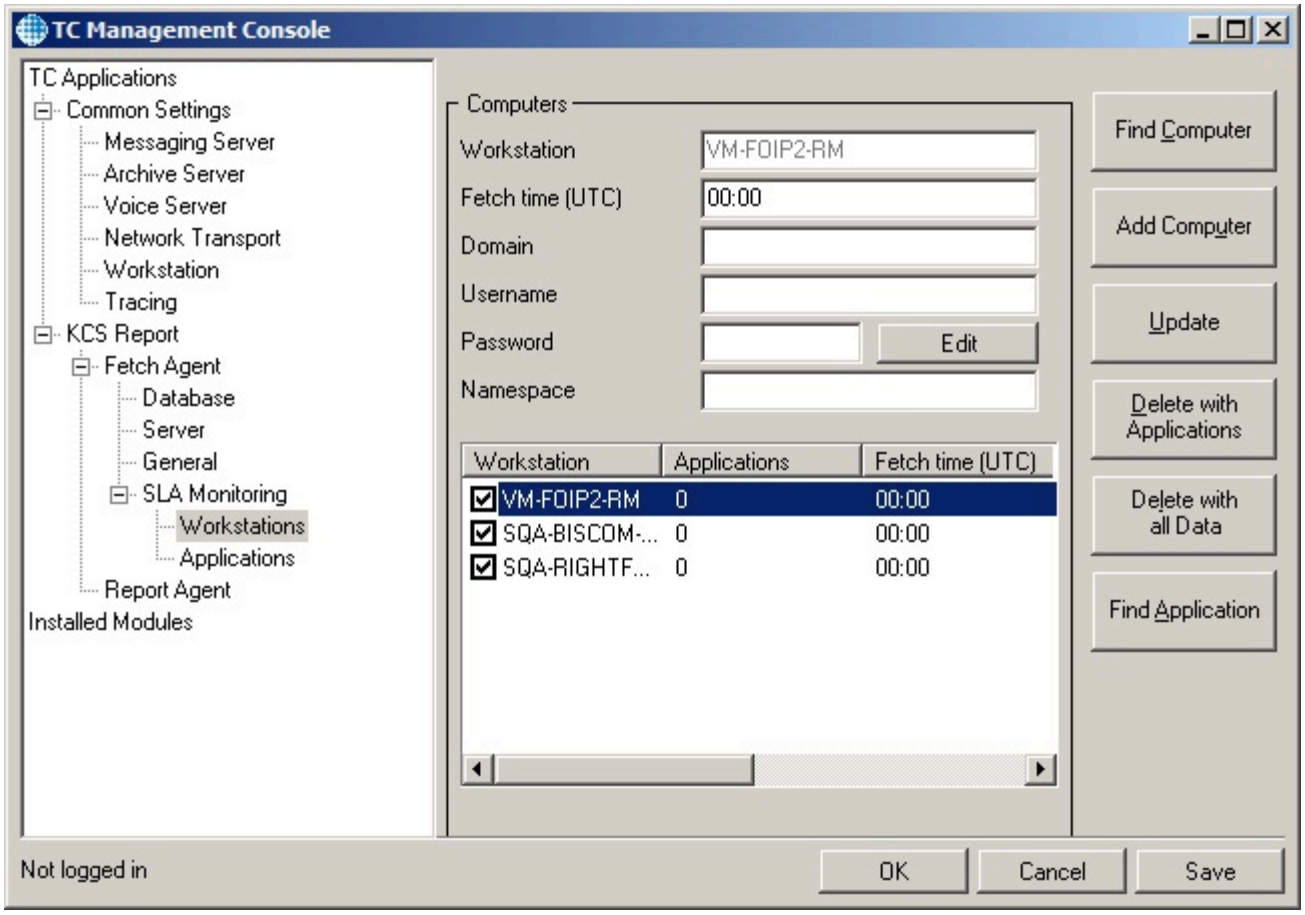
The last error that occurred is also displayed as a tool tip when the mouse cursor is over the list of domains.

If TC/Report runs on a Windows Server 2008 (or later) computer and you see no other computers in the network neighborhood, make sure that Network Discovery is enabled (Control Panel | Network and Sharing Center) and the local Computer Browser service is running.

Select all computers that hold KCS applications you want to monitor (e.g. TCOSS servers, link servers, voice servers). When you finished your selection, click OK to add these computers to the workstations panel.

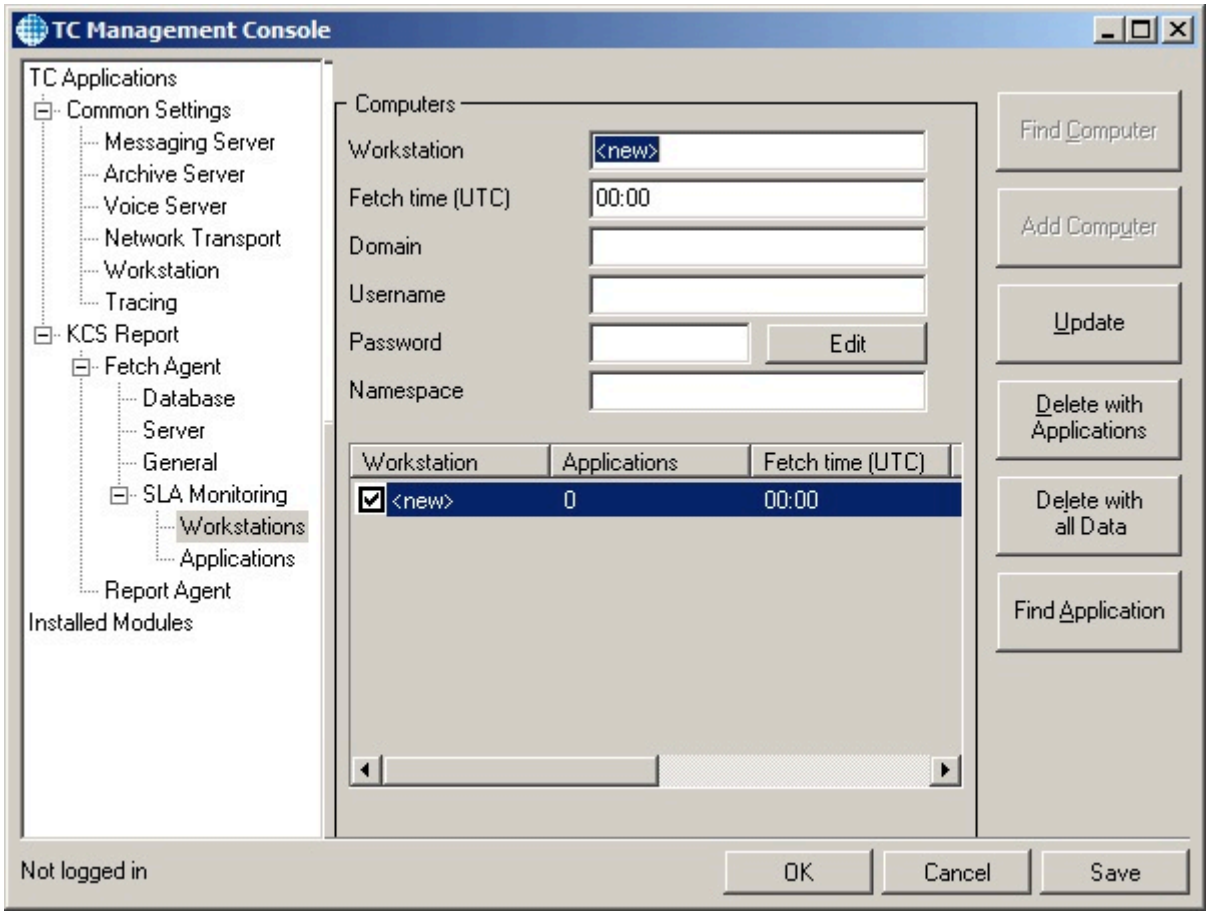


A message box shows how many workstations have been added. If you selected a computer that is already in the list of workstations, it is not added and is counted as a “duplicate workstation” in this message box.



As an alternative to the **Find Computer** button, you can also add a single workstation via the button **Add Computer**.

Add Computer creates a new entry in the workstation list and allows you to edit the workstation name, as well as all other settings on the panel. Once you click **Update**, the workstation name cannot be edited any more.



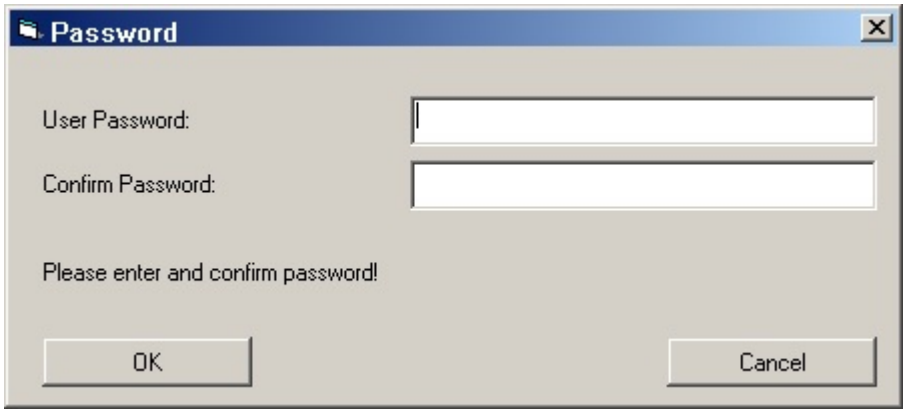
Configuring access to monitored computers

Important If Windows Firewall is active on the remote computer, the Windows Firewall inbound rule “Remote Administration (RPC)” must be enabled. Otherwise, TC/Report cannot access the remote event log.

The workstations panel now shows the names of the selected workstations. For every workstation, you must configure at which time of the day TC/Report shall fetch event log entries. TC Management Console suggests that fetching occurs at midnight (00:00). You can change the fetch time of the selected workstation by editing the field **Fetch Time (UTC)**.

In most cases, you have to configure the values in columns **Domain**, **Username** and **Password**. These values define which credentials the process FetchWorkstations.exe uses when reading the event log of the workstation. If you leave them empty, FetchWorkstations.exe uses its own process user account.

To edit the password, click **Edit**. The **Password** dialog is opened. For security, the password has to be entered twice.



On Windows Server operating systems, only administrators can read the application event log.

If the remote computer runs Windows Server 2008 or later with UAC (User Access Control) enabled, there are further restrictions:

- Apart from the built-in computer administrator, only administrators that are domain users can connect.
- If the remote computer is part of a workgroup and has UAC enabled, the only way to enable event log fetching is by disabling remote UAC. On Windows Server 2008, create the following REG_DWORD registry value and set its value to 1: HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\system\LocalAccountTokenFilterPolicy

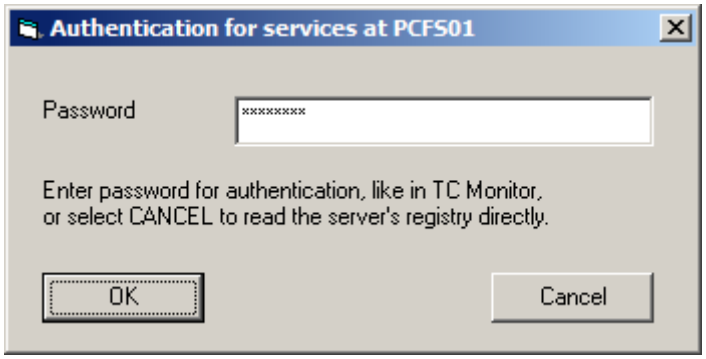
The **Namespace** field can usually be left empty. It must be configured only for computers where the default WMI namespace has been changed via *Computer Management* | *Services and Applications* | *WMI Control* or by editing the registry value *HKLM\Software\Microsoft\WBEM\Scripting\Default Namespace*. In this case, you should set it to “root\cimv2”.

Finding installed applications

TC Management Console can find installed KCS applications automatically. Activate the checkboxes beside the workstation names where you want to find installed KCS applications. Then click **Find Application**.

TC Management Console uses several methods for this purpose:

First, it tries to connect to the TCSRV service on the remote workstation. The first connection attempt is done via Named Pipes, if this fails, the TCP/IP protocol is used. If the TCSRV service uses password protection, the user has to enter the password:



If the connection to the remote TCSRV service succeeds, TC Management Console retrieves process information from TCSRV. Otherwise, it tries to connect directly to the remote registry.

Prerequisites:

To retrieve the application list from TCSRV, the remote TCSRV service must run.

The remote TCSRV can be configured to use Named Pipes or TCP/IP for monitoring. This is configured in registry value HKEY_LOCAL_MACHINE\Software\TOPCALL\Boot\TCRPCServerMode:

3 = Named Pipes (default)
1 = TCP/IP

If TCP/IP is used, port 64385 on the remote server must not be blocked. Apart from this, no additional restrictions apply, i.e. it does not matter which user is logged in.

If TCSRV uses Named Pipes for monitoring, or if direct registry access is necessary (e.g. when TCSRV does not run), “Find Application” works only if the user who started TC Management Console has administrator rights on the remote machine: You must be logged on as a domain user who is member of the remote machine’s Administrators group, or you must connect to the remote computer explicitly, using one of the remote computer’s administrator accounts (via Start | Run | \\machine-name).

Issues with Windows Server 2008 and later:

Authentication on the remote computer becomes even more difficult if UAC enabled Windows Server 2008 (or later) are involved:
– If TC Management Console is started “as administrator” on these operating systems, connecting to a remote computer is ineffective (because the process uses the built-in administrator user instead of the interactively logged in user).

– If the computer to be monitored runs Windows Server 2008 or later, connecting with a local administrator account is only possible if remote UAC is disabled on the remote computer, via the registry setting:

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\system\
 LocalAccountTokenFilterPolicy=1
```

1) To grant individual users remote access to the registry

- Open (or create) the following registry key:
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg
- Add the following value:

Value Name: Description
Data Type: REG_SZ
String: Registry Server

- Select HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg and click Permissions. Add users or groups to which you want to grant access.
- Exit Registry Editor and restart Windows.

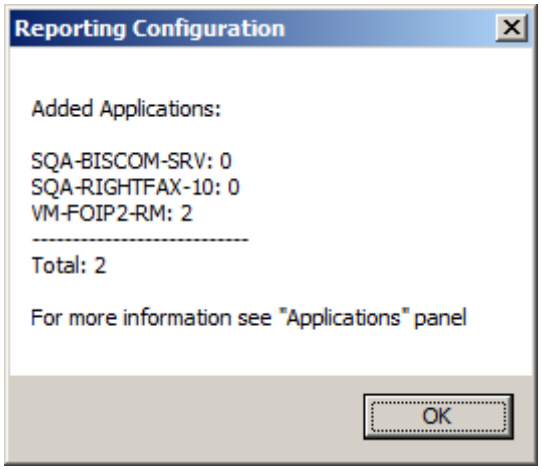
2) To bypass access restriction to the HKEY_LOCAL_MACHINE\Software\Topcall key:

You can configure Windows to bypass the access restriction to certain keys by listing them in the Machine value under the AllowedPaths key.

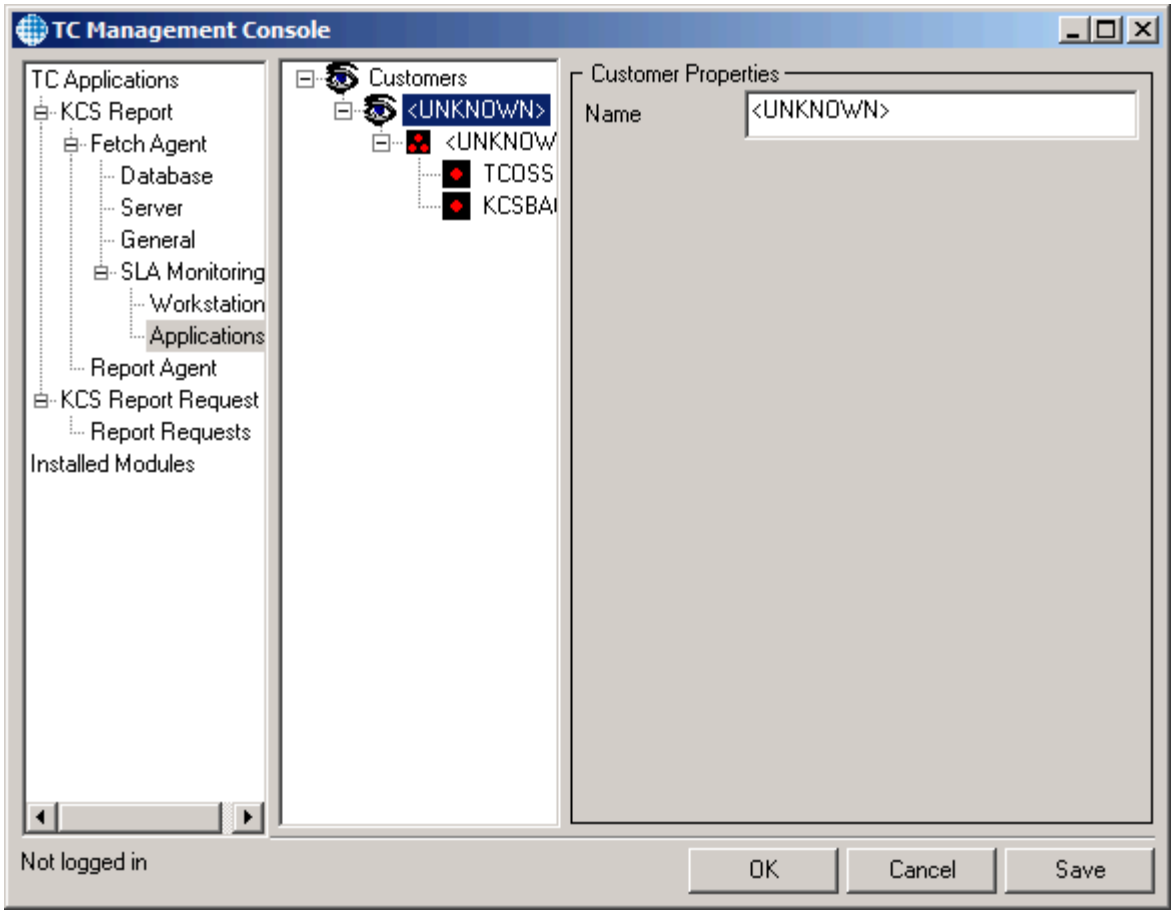
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg\AllowedPaths

This is a REG_MULTI_SZ value. Just add Software\Topcall to this value.

Found Applications



A message box shows how many applications have been added to the database. You can assign these applications to customers and application groups in the Applications panel.

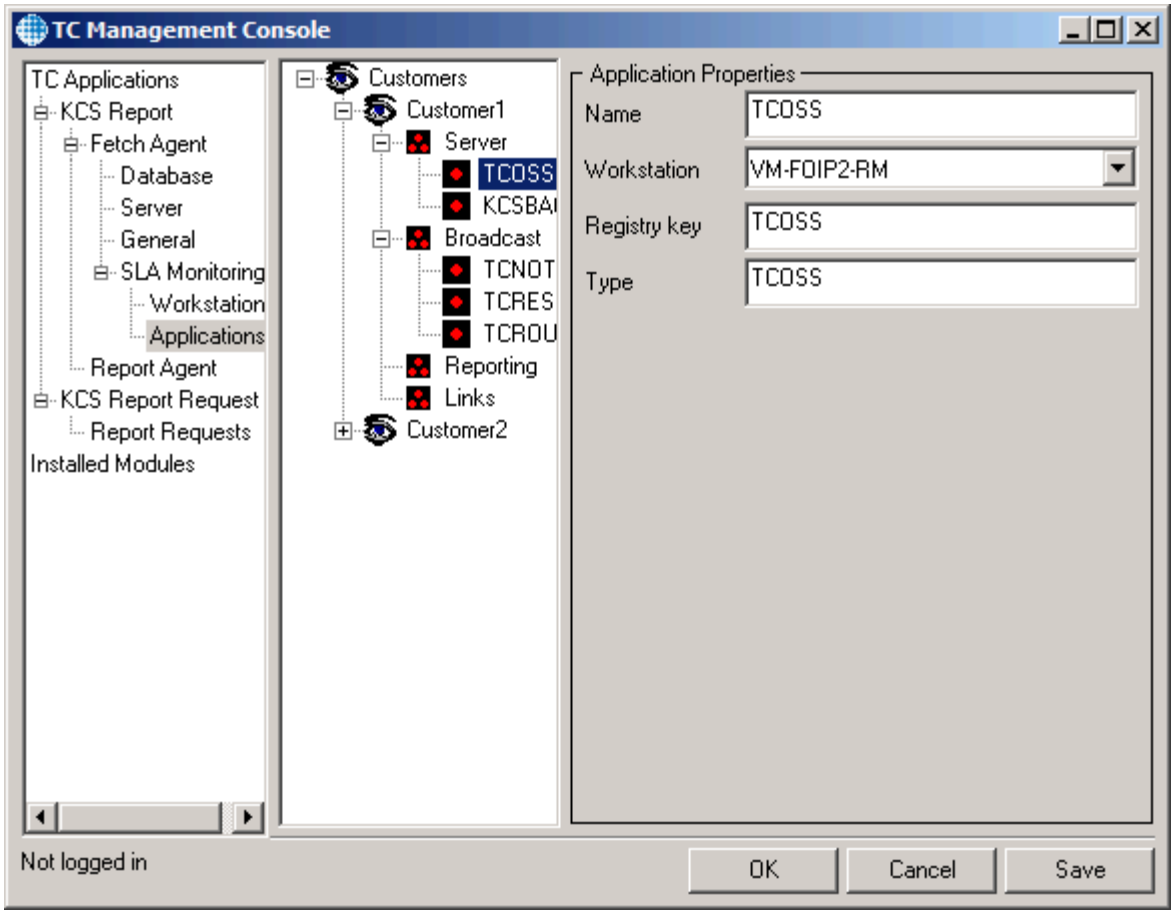


The **Applications** panel shows a hierarchy of customers, application groups and applications. Every customer can have several application groups and every application group can hold several application instances.

This hierarchical organization allows to create application downtime reports that are categorized by customer and application group, - correctly assigning applications therefore makes the report more readable.

The automatic lookup of applications detects the customer name of TCOSS instances, but assigns most other applications to customer <UNKNOWN>. It also places all applications of a customer into 1 application group called <UNKNOWN>. It is now up to the administrator to assign applications to application groups and customers correctly. The applications tree view has context menu items that allow you to create application groups and applications, to copy and paste applications, to move them to a different group via drag and drop, and to change names and descriptions of the items.

Here is an example of an application hierarchy:



In this configuration, every customer has the same set of application groups: Server, Broadcast, Reporting etc. The administrator created these groups manually below the customer nodes.

Every application group holds 1 or more applications that perform the same work (e.g. primary and secondary TCOSS server) or work together for the same purpose (e.g. the TCBROADCAST server applications).

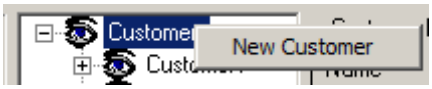
Some of these applications only run in 1 instance, nevertheless the database holds a separate copy of the application for every customer (e.g. the Broadcast applications, the Reporting applications and TCDCLINK). The additional copies were made manually via context menu items *Copy* and *Paste*.

With this configuration, it is easy to create a report that shows times of unavailability for all applications used by a single customer, or even only for an application group of this customer.

As an alternative, you could assign applications used by several customers to a pseudo customer “All”.

Creating a new customer definition

To create a new customer container, right click the Customers icon and choose context menu item “New Customer”.



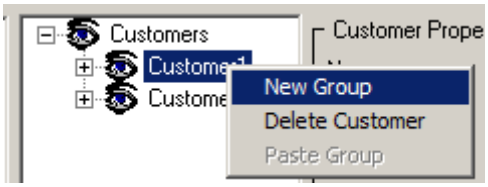
Please note that customer definitions without associated applications will not be stored in the database.

Deleting a customer and all associated application groups and applications

To delete a customer definition, right click the customer icon and choose context menu “Delete Customer”.

Creating a new application group

To create a new application group, right click a customer icon and choose context menu item “New Group”.



Changing application group properties

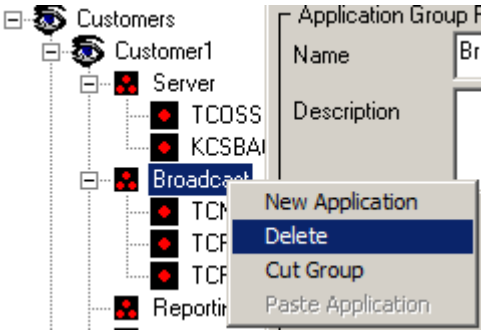
When an application group is selected in the tree view, you can edit the following settings on the right side of the window:

Name: a descriptive name for the application group.

Description: a longer description of the application group.

Deleting an application group

To delete an application group, right click the application group icon and choose context menu item “Delete”.



Creating a new application

It is possible (but usually not necessary) to create an application object manually, by right clicking an application group icon and choosing context menu item “New Application”.

Changing application properties

When an application is selected in the tree view, you can edit the following settings on the right side of the window:

Name: a descriptive name for the application.

Workstation: This combo box holds the name of the workstation where the application runs. For application objects that were created automatically, this field is filled correctly and need not be changed. For application objects you created manually, choose one of the configured workstations.

Registry key: the registry subkey of the application (below HKLM\Software\Topcall). For application objects that were created automatically, this field is filled correctly and need not be changed. For application objects you created manually, enter the registry key of the application.

Type: the application type. For application objects that were created automatically, this field is filled correctly and must not be changed. For application objects you created manually, enter the correct type from the table in section “Application types”.
The table defines application types for standard KCS applications. For all non-standard applications, e.g. created by Professional Services, use application type “OTHERS”.

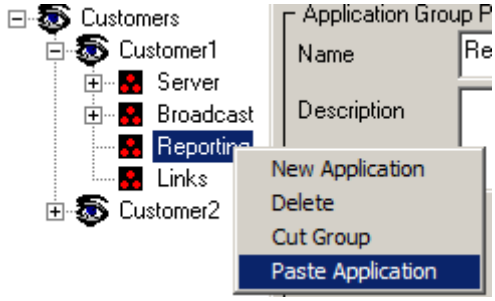
Deleting an application

To delete an application, right click the application icon and choose context menu item “Delete”.

Creating a new copy of an application

For applications that are used by several customers, like TC/Report or the TC/LINK document converter, you will want to copy the application object to every customer container.

Right click the application icon and choose context menu item “Copy”. Then right click the application group icon where the new copy shall be created and choose context menu item “Paste Application”.



Moving an application to a different application group

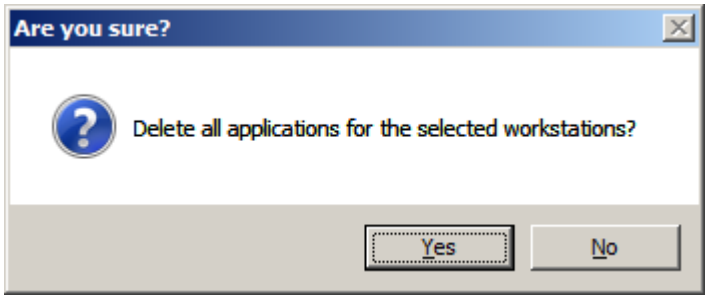
This can be done by simply dragging the application icon to its new position.

As an alternative, you can use context menu items “Cut” from the application object and “Paste Application” from the application group object.

Deleting a workstation

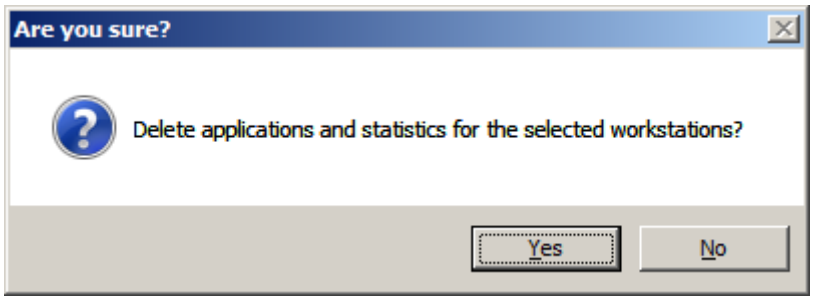
Button **Delete with Applications** in the Workstations panel allows deletes the selected workstations and all applications for these workstations. Downtime information still remains in the database.

A message box gives you the option to cancel the action:



Button **Delete with all Data** in the Workstations panel deletes the selected workstations, all applications and application statistics for these workstations. Downtime information is also deleted.

A message box gives you the option to cancel the action



Please note that (different from previous versions), the database changes are only done when either **OK** or **Save** button are clicked.

Application types

The TCREPORT database holds information about events and registry keys of the following application types. Only these types can be detected automatically.

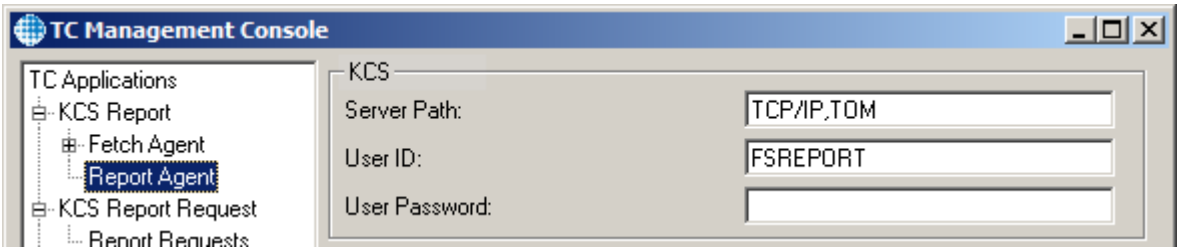
Name	Description	Needs TCSRV start event (Boot\ProcessReady)
TCDC_LINK	Link based document converter	YES
TCDC_TCOSS	TCOSS based document converter	NO
TCREPORT_FETCH	TC/Report Fetch Agent	YES
TCREPORT_REPORT	TC/Report Report Agent	YES
TCOSS	TCOSS instance	NO

Name	Description	Needs TCSRV start event (Boot\ProcessReady)
TCARCH	KCS Archive	NO
TCSTATUS	TCSTATUS	NO
TCECP	Voicemail, Voicelink	YES
TCPOP3	POP3 server extension	YES
TCIMAP	IMAP	YES
TCLDAP		YES
TCFILBRK	File Break	YES
TCJUKE		YES
TCLANPRT	Lan Print	YES
TCMSGWAIT		YES
TCWEB	TCWEB	YES
TCMWAMX	Message Wait Agent for Exchange	NO
TCLINKAC	TC/LINK-AC	NO
TCLINKSJ	TC/LINK-SJ	NO
TCLINKMQ	TC/LINK-MQ	NO
TCLINKGW	TC/LINK-GW	NO
TCLINKX4	TC/LINK-X4	NO
TCLINKFI	TC/LINK-FI or TC/LINK-CCD	NO
TCLINKLN	TC/LINK-LN	NO
TCLINKMD	TC/LINK-MD	NO
TCLINKWM	TC/LINK-WM	NO
TCLINKSM	TC/LINK-SM or TC/LINK-OC or TC/LINK-SDD or TC/LINK-MFP	NO
TCLINKSC	TC/LINK-SC	NO
TCLINKSI	TC/LINK-SI	NO
TCLINKMX	TC/LINK-MX	NO
TCROUTE	TCROUTE process (part of TC/BROADCAST)	YES
TCOCRMF	TCOCRMF process (part of TC/BROADCAST)	YES
TCNOTIF	TCNOTIF process (part of TC/BROADCAST)	YES
TCRESMF	TCRESMF process (part of TC/BROADCAST)	YES
TCPROBE	Probe Agent	NO
TCFAXFORM	OFR detection (custom part of TC/BROADCAST)	YES
TCMFFORMSGEN	Coversheet generation (custom part of TC/BROADCAST)	YES

Name	Description	Needs TCSRV start event (Boot\ProcessReady)
TCSOAP	TC/SOAP	YES
OTHERS	Any other application started by TCSRV	YES

Report Agent Configuration

Parameters used for the Report Agent:

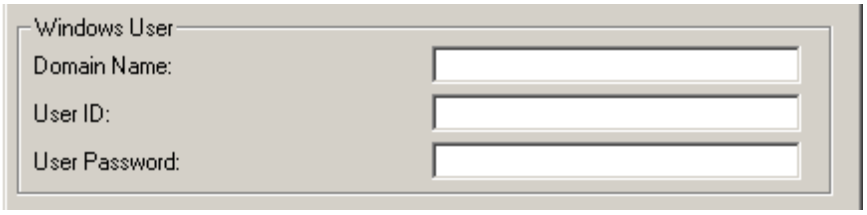


Server Path (for a model 22x it can be an alternative path, for **Example**: TCP/IP,PRIMARY|TCP/IP,SECONDARY)

User ID of user whose inbox is polled.

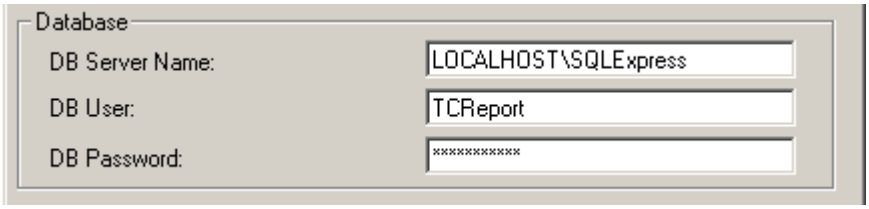
User Password of user to be connected to the TCOSS server.

Parameters for TCSRV:



If printing directly from TC/REPORT, a Windows user, domain and password has to be supplied

Parameters for the database:



DB Server Name – Name of the MS SQL Server.

DB User – Name of the database user created when installing the Fetch agent.

DB Password – Password of this user

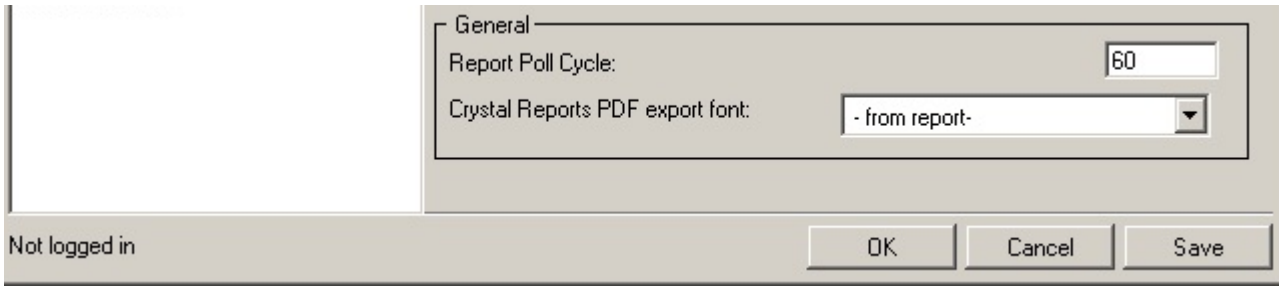
General Parameters:

Report Poll Cycle

This number (seconds) defines how often the server is polled for new command messages (report requests).

Crystal Reports PDF export font

In an international system, you should explicitly configure the font used by the Crystal Reports runtime library for generation of PDF reports. The default font from the standard reports does not support the full range of UTF-16 characters. The combo box offers all fonts installed on the local system. Take care that font names starting with @ indicate vertical fonts (each character rotated by 90 degrees).



Chapter 6

Special Features of the Fetch Agent

This chapter describes the special features of Fetch Agent.

Handling of Media Servers in ASP Systems

With ASP systems, different media servers can have different channel assignments. For example, channel 05 on MEDIASERVER1 may be a fax channel, whereas channel 05 on MEDIASERVER2 is a telex channel. This scenario could not be handled correctly with older TCReport versions.

Now, the media servers must be configured as separate, disabled servers. Additionally, there is a new checkbox **Media Server** in the TC Management Console server list. Make sure this checkbox is marked for media servers, otherwise they will show up in report headers when requesting reports for all servers.

In the above example, configure 2 disabled servers MEDIASERVER1 and MEDIASERVER2. Select MEDIASERVER1 and use the **HW Channels** button to load the channel configuration from the media server. Then do the same for MEDIASERVER2.

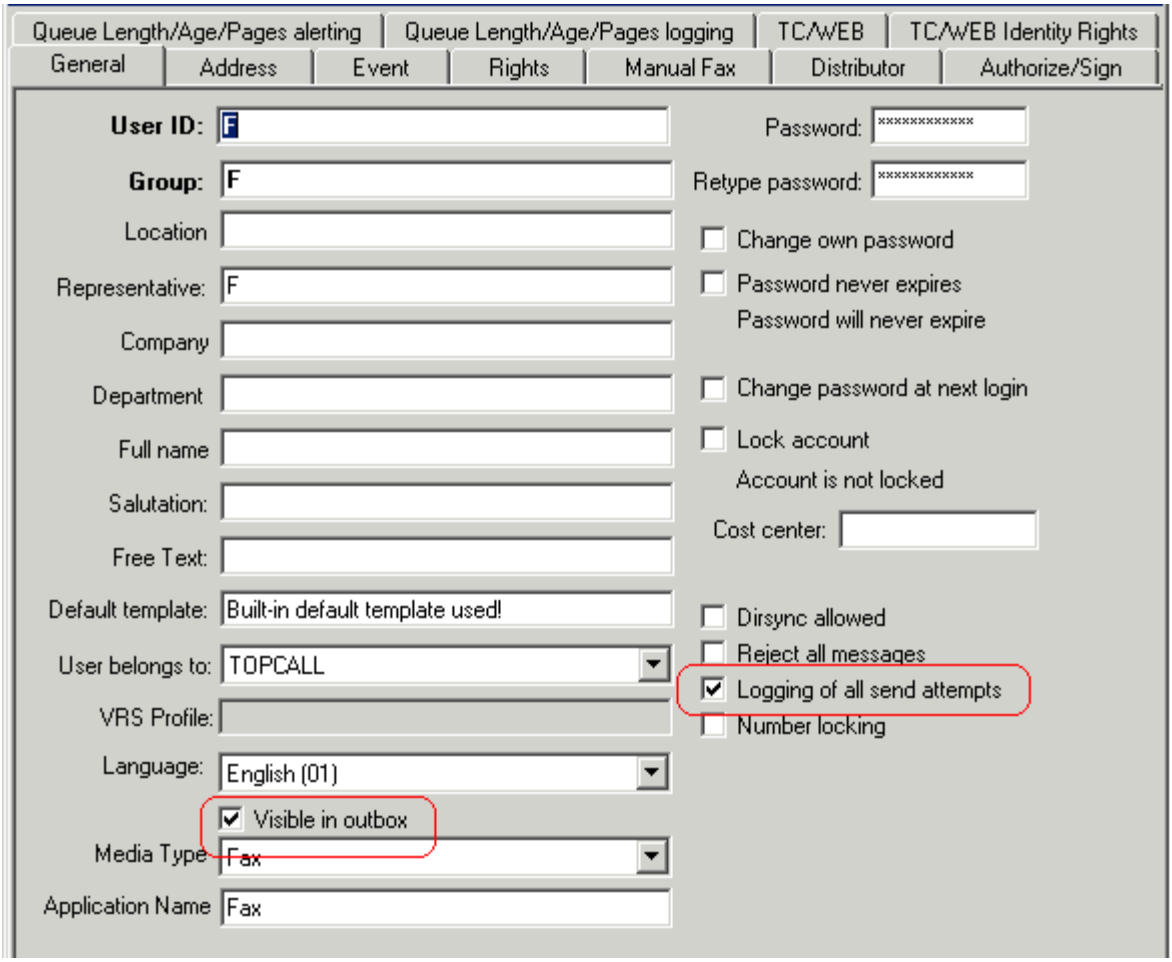
When processing a send order that passed over a media server, the Fetch Agent and the reports created by the Report Agent check the channel configuration for the media server.

Handling of LCR (Least Cost Routing) Systems

This section explains how to configure TCOSS and TC/Report in order to get correct line statistic reports for least cost routing systems (i.e. showing channel usage for the node that sent or received the fax or telex).

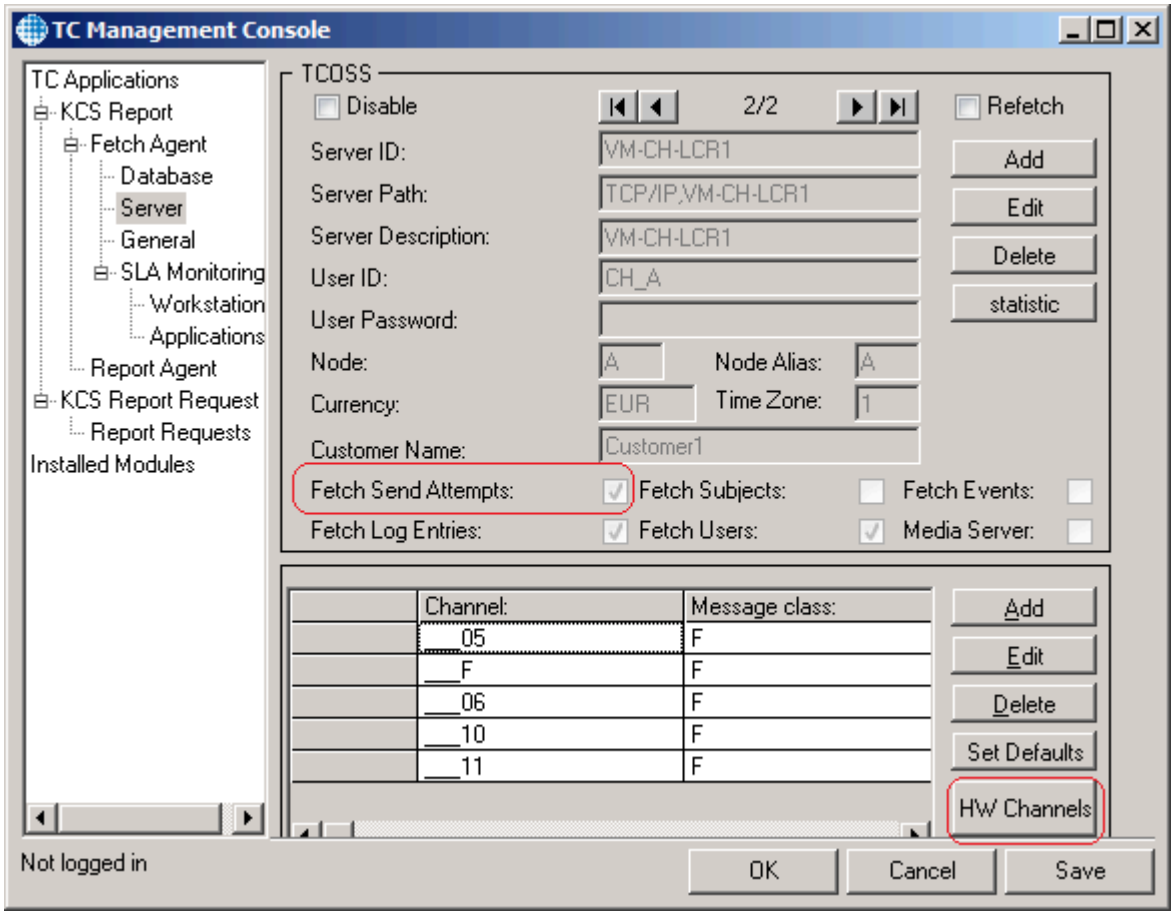
TCOSS configuration:

If there are dedicated KCS users defined for fax or telex channels (or channel groups), e.g. the ‘F’ user for fax, you should make sure that these user profiles have the flags “Visible in outbox” and “Logging of all send attempts” enabled.



TC/Report configuration:

In the TC Management Console panel FetchAgent | Servers, configure a server for every node of the LCR system. Enable the option “Fetch Send Attempts” for all of them, and use the “HW Channels” button to import channel information from TCOSS.



Handling of Incoming Fax or Telex

In previous releases of the Fetch agent, some types of messages were treated as incoming fax (or telex), although they were not. For example, the distribution of a fax by an operator was counted as an additional incoming fax in UNIV and LINES report. In-events to mail system accounts were also regarded as incoming faxes.

Now, the fields TS_REC_CHANNEL, TS_REC_SERV_ID, Chg_In and Cc_In in the Action_Table are empty for all event and notification messages, and also for messages created by a distributor.

As a consequence, only real incoming fax and telex messages (with channels configured in the Channel_Table) are displayed in the LINES and TCLINESN reports.

If you use the above fields in a UNIV report, all send orders that are not incoming fax or telex will show up as “OTHER” for these categories.

The field MsgClass_Originator is treated differently: The originator message class stays “FAX” for in-events and distributed faxes. This field tells you about the message class of the original send order

that triggered this send order. It can be used, e.g. to count incoming faxes forwarded to Outlook users (MsgClass_Originator = “FAX”, MsgClass_Recipient = “EXCHANGE”).

Automatic Deletion of Oldest Records

The database usually consists of two files, the actual database file and a log file (TCREPORT.MDF and TCREPORT_LOG.LDF), which stores all the database actions. When part of the database is deleted, this is recorded in the log file, which makes the latter grow, so the database will first grow even more before shrinking again beneath the maximum size. That is why the hard disk requirements are double the maximum database size.

KCS Setup creates the TCREPORT database with the following size values:

File Name	Initial size (MB)	Maximum file size	Remark
TCREPORT	10	Unlimited	Data file
TCREPORT_log	10	Unlimited	Transaction Log

SQL server interprets the initial data file size as a minimum size for this file. This means that it is not possible to shrink the file to a smaller size.

In the same way, it is possible to define a maximum size for the data file, instead of unrestricted growth.

In addition to these file size limits, TC/Report has a maximum size for the database, configurable via the TC/Management Console. You can configure either the maximum physical size of the database file (for a database with automatic growth) or the maximum size of the database content (for a database with fixed size).

When the maximum size is reached, the Fetch Agent deletes the oldest 10% of the table entries and repeats this until the database size is below the maximum again. For database files that are allowed to grow automatically, the Fetch Agent then tries to shrink the database file to the smallest possible size.

Note You can disable TC/Report’s maximum database size check by setting this limit to 0.

To avoid conflicts between the file size limits (that may be changed by an SQL operator) and the database size limit managed by the Fetch Agent, TC/Report ensures that the configured maximum database size (registry: DBMaxSize) is either 0 (disabled) or is within the following limits:

- Lower limit: (sum of database file minimum sizes) * 1.05
- Upper limit (if file growth is restricted for all database files): sum of database file maximum sizes
- Upper limit (if one or more database files are unlimited in growth): unlimited

TC Management Console reads the limits from the database and adjusts the valid maximum database size that is displayed. If you want to store an invalid value, a message box will pop up, with a text as follows:

Due to database size restrictions, value DBMaxSize is changed from XXX to YYY.

The adapted value (YYY) is stored in the TC/Report configuration.

TC/Report Fetch Agent also reads the limits from the database and adjusts the *DBMaxSize* registry value if necessary. This is done after startup, and then once per hour and whenever the configured *DBMaxSize* limit is reached (before deleting the oldest entries).

If TC/Report Fetch Agent changes the *DBMaxSize* value, it will write the following informational message to the application event log:

ID:8557 Due to database size restrictions, value DBMaxSize is changed from XXX to YYY.

Unfortunately, this mechanism does not handle the overall database size restriction in SQLExpress (4 GB).

Backup of the Database

The Fetch Agent stores every entry in its database. The database name is configurable, with default settings the database file is named „TCReport.mdf“. When the maximum size is reached the fetch agent deletes the oldest entries. To avoid to lose data the fetch agent can create backup databases automatically (provided that the SQL user has the right to create databases, - otherwise the backup databases must be created manually). These databases are stored in the same directory as the default database; with default settings, they are named TCReportBak00001, TCReportBak00002, etc.

The backup databases contain the following tables:
Action_Table
Log_Table
User_Table
Server_Table

(plus the Probe_Table if the TC/PROBE is connected to this database).

As a rule, the tables in the backup database contain only records that relevant for the time period covered by the backup.

In previous versions, backup databases were written record by record during the Fetch process. This made backups for more than 1 server impossible. Now, a backup in 1 step, as soon as the required information is available for all enabled servers.

The maximum number of backup databases and the maximum size can be configured in the Management Console; *(see Fetch Agent Configuration, section Database Property Page).*

It is also possible to configure the time period of the stored entries. E.g. if you want to have one backup database for each month the first database contains entries of January, the second database entries of February, etc.

The information on which time period is stored in the backup databases is stored in the Database_table (in the default database TCReport.mdf).

When a backup database is overwritten, either because only one backup file was configured or because the maximum number of backup files was created and the first one needs to be overwritten, the fetch agent sets the attribute “deleted” in the Database_table.

The attribute “ready” in the Database_table means that the maximum size of this backup database is reached or the time period of this backup database is fetched completely. This means no more entries will be stored in this database and it is “ready” to be stored on a storage media, if so desired.

Example:

Maximum number of backup databases: 5

Create backup databases periodically: each month (you can also define other values here, such as day, year, week, etc.)

DB_ID	filename	datefrom	dateto	ready	deleted
1	TCReportBak00001	01-Jan-2000	31-Jan-2000 23:59:59	True	True
2	TCReportBak00002	01-Feb-2000	29-Feb-2000 23:59:59	True	True
3	TCReportBak00003	01-Mar-2000	31-Mar-2000 23:59:59	True	True
4	TCReportBak00004	01-Apr-2000	30-Apr-2000 23:59:59	True	True
5	TCReportBak00005	01-May-2000	31-May-2000 23:59:59	True	True
6	TCReportBak00001	01-Jun-2000	30-Jun-2000 23:59:59	True	True
7	TCReportBak00002	01-Jul-2000	31-Jul-2000 23:59:59	True	True
8	TCReportBak00003	01-Aug-2000	31-Aug-2000 23:59:59	True	False
9	TCReportBak00004	01-Sep-2000	30-Sep-2000 23:59:59	True	False
10	TCReportBak00005	01-Oct-2000	31-Oct-2000 23:59:59	True	False
11	TCReportBak00001	01-Nov-2000	30-Nov-2000 23:59:59	True	False

When a database is done, the fetch agent creates a new database if more than one database was configured, with the next number. The fetch agent can be configured to send a message to a KCS user (you must specify the KCS internal service and the user; e.g. “TOPCALL,USER”) or start a batch file (specify the full path; configured in TC/Management Console) e.g. to inform an administrator that the backup is ready to be stored on a storage media.

When you request a report and the requested time period is not in the default database (TCReport), the report agent looks in the Database_table to find out in which database the requested time period is. The report will be created from the backup database.

It is also possible to specify the database-name explicitly in a parameter: Database = …

In this case the report agent does not check if the requested time period is content of the requested database.

Calling a Batch File When Backup Is Ready

The batch file gets the following parameter (you get the values of the parameter with the placeholder %1, %2, %3):

%1 = Database name (e.g. TCReportBak00001)

%2 = DateFrom in the format yyyymmdd-hhmmss (the beginning of the time period; e.g. 20000101-000000)

%3 = DateTo int the format yyyymmdd-hhmmss (the end of the time period; e.g. 20000131-235959)

Example for the batch file, that is started automatically:

```
osql -U sa -P password -Q " EXEC sp_addumpdevice 'disk', 'device_%1', 'c:\mssql7\backup
\%1.dat'" >>c:\logfile.txt
echo database %1 >>c:\logfile.txt
echo datefrom %2 >>c:\logfile.txt
echo dateto %3 >>c:\logfile.txt
osql -U sa -P password -Q "backup database %1 to device_%1" >>c:\logfile.txt
echo ----- >>c:\logfile.txt
rem Note: here you can copy the file c:\mssql7\backup\%1.dat to a storage media like CD
 or tape.
```

Note

- You cannot copy the database files (TCReportBak00001.mdf and TCReportBak00001_log.ldf) directly because they are locked by the SQL server. This example prepares a file which is not locked and can be copied (c:\mssql7\backup\%1.dat).
- The current directory of the batch file is c:\topcall\shared. You should specify a path when you specify any filename in the batch file.
- This example uses the stored procedure of the SQL server “sp_addumpdevice”. This procedure needs a database user with administrator rights (like the database user ‘sa’).
- This stored procedure should be called only once for each backup database otherwise it will fail. It would be best not to make it part of the batch file. If you have it in the batch file however, it will result in an error message in the log file, but would not hinder the rest of the batch file from being executed correctly.

Sending a Message When Backup Is Ready

The message that TC/Report sends when a backup has been created contains the following information:

- Name of the backup database
- The time frame for this backup
- The creation date of the backup

Date and time values are in typical TCOSS format (YYYYMMDD-hhmmss).

Example:

```
*******************************************************************************
                                    TC/REPORT
Backup "TCR1Bak00002" (20060424-000000 - 20060430-235959, created 20060614) is ready to
 be stored on a storage media.
*******************************************************************************
```

Requirements and Restrictions of the Backup

The backup database is written in one step, as soon as one of the following conditions applies:

- Time limit reached (the time frame for this backup is fetched for all enabled servers)
  Note: If a disabled server is activated after the backup is made, these newly fetched entries are not inserted to existing backup databases.
- Size limit reached (presumable size is calculated, may be a bit incorrect)

- The main database reaches its limit and deletion of the oldest 10 % of entries would delete information not yet stored in a backup.

The last condition (main database full) may lead to backups that hold only part of the configured time period and are smaller than the configured maximum backup size.

Recommendation: The maximum size of the default database should be greater or equal to the maximum size of the backup, otherwise the backup will be truncated into small portions or will not hold all information.

For more than one TCOSS instance:

- With the option **send a message to user**, the fetch agent uses the first enabled TCOSS server. Make sure that the configured user exists on this server.

For requesting a report where the time period is in a backup database:

- The requested time period must be in only one database, otherwise you will get the error message **can't access more than one database**. You have to split the time period of the report request into various requests.

Basic Integration with UTC-Based TCOSS

The Fetch Agent now uses a dedicated TCSI application session for every TCOSS connection. Together with the configuration changes described below, this change enables integration with UTC-based TCOSS instances.

As described in the TCOSS Release Documentation version 7.57.03, it is possible to run TCOSS instances with UTC time or with a defined offset from UTC time. The first case is used for new installations, whereas the second case is used when updating existing installations.

Message-related time stamps in the TC/Report database still are based on the TCOSS time zone (UTC or UTC plus offset). These time stamps are the basis for creating backups and reports.

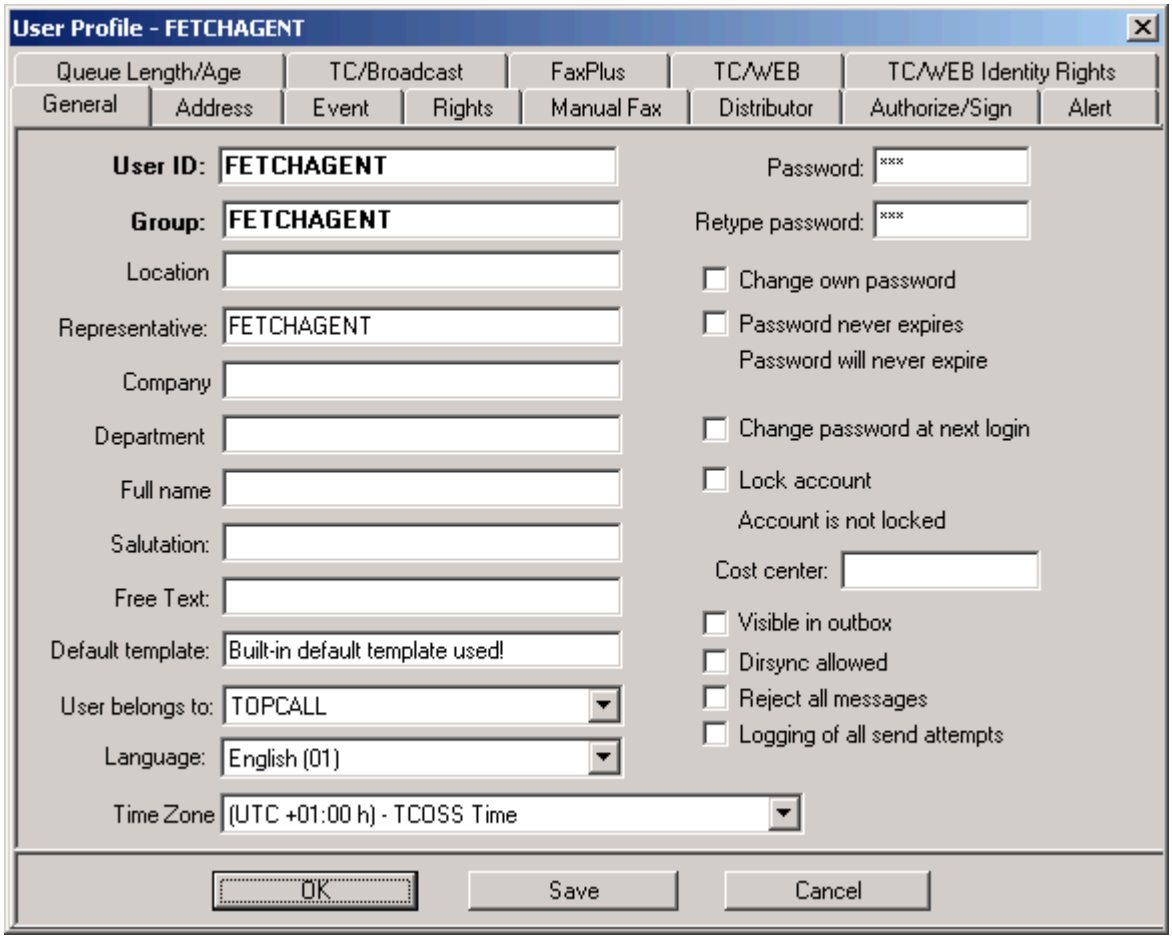
Configuration changes: User Profiles

Both agents use KCS user profiles. For the Fetch Agent, the user name is stored in the Server_Table (field *User_ID*). For the Report Agent, the user name is stored in the registry (*Topcall\UserID*).

For integration with UTC-based TCOSS instances, a time zone must be defined for these user profiles:

- Either the TCOSS time zone (for updated TCOSS installations, where an offset to UTC time is defined).
- Or the UTC time zone (for new TCOSS installations without an offset).

To configure the time zone, open the user profile with TCFW version 5.12.00 or above.
In the Time Zone list box, select TCOSS time or UTC time.



Configuration changes: Registry

For Fetch Agent and Report Agent, create a REG_DWORD registry value *TCSI\AutoTimeZone* and set it to 1.

This enables automatic usage of the time zone configured for the agent's KCS user.

Chapter 7

Maintenance

This chapter describes the maintenance details for Fetch Agent.

Performance Counters

A few performance counters have been implemented for the Fetch Agent and for the Report Agent. These counters may be used e.g. for trending analysis.
You must explicitly enable the counters by setting *EnablePerformanceCounters* to 1 below the registry keys of the agents.

Fetch Agent: global counters

Counter Name	Object Name	Instance Name	Description
IsDeleting	TCReport_Fetch_General	TCREPORT_Fetch	Is 1 while deleting old records, otherwise 0.
Entries/day	TCReport_Fetch_General	TCREPORT_Fetch	Estimated number of fetched entries per day (based on last 10 poll cycles)
Entries/hour	TCReport_Fetch_General	TCREPORT_Fetch	Estimated number of fetched entries per hour (based on last 10 poll cycles)
Entries/min	TCReport_Fetch_General	TCREPORT_Fetch	Estimated number of fetched entries per minute (based on last 10 poll cycles)
Entries/sec	TCReport_Fetch_General	TCREPORT_Fetch	Average number of fetched entries per seconds (based on last 10 poll cycles)
Users/day	TCReport_Fetch_General	TCREPORT_Fetch	Estimated number of fetched users per day (based on last 10 poll cycles)
Users/hour	TCReport_Fetch_General	TCREPORT_Fetch	Estimated number of fetched users per hour (based on last 10 poll cycles)
Users/min	TCReport_Fetch_General	TCREPORT_Fetch	Estimated number of fetched users per minute (based on last 10 poll cycles)
Users/sec	TCReport_Fetch_General	TCREPORT_Fetch	Average number of fetched users per second (based on last 10 poll cycles)

FetchAgent: per server counters

Additional counters (similar to the general Entries/xx and Users/xx counters) allow to monitor the number of entries or users fetched from a single TCOSS instance. The counter values are calculated as: (number of entries fetched from server during last 10 poll cycles) / (duration of last 10 poll cycles).
Thus, the values are also influenced by the performance of the other servers that are polled.

Only the Entries/sec are the “real” values, the other values correspond to this one e.g. entries/min = entries/sec * 60.

Disabled servers are also part of the performance counters, the values are always 0.

The counters “Fetchtime/Entry(msec)” and “Fetchtime/User(msec)” measure the net duration of fetching short term archive entries or user entries from a single TCOSS instance. These counters are not influenced by the performance of the other servers.

Counter Name	Object Name	Instance Name	Description
Entries/day	TCReport_Servers	Server name	Estimated number of fetched entries per day (based on last 10 poll cycles)
Entries/hour	TCReport_Servers	Server name	Estimated number of fetched entries per hour (based on last 10 poll cycles)
Entries/min	TCReport_Servers	Server name	Estimated number of fetched entries per minute (based on last 10 poll cycles)
Entries/sec	TCReport_Servers	Server name	Average number of fetched entries per seconds (based on last 10 poll cycles)
Users/day	TCReport_Servers	Server name	Estimated number of fetched users per day (based on last 10 poll cycles)
Users/hour	TCReport_Servers	Server name	Estimated number of fetched users per hour (based on last 10 poll cycles)
Users/min	TCReport_Servers	Server name	Estimated number of fetched users per minute (based on last 10 poll cycles)
Users/sec	TCReport_Servers	Server name	Average number of fetched users per second (based on last 10 poll cycles)
Fetchtime/ Entry(msec)	TCReport_Servers	Server name	Net duration of fetching one entry.
Fetchtime/ User(msec)	TCReport_Servers	Server name	Net duration of fetching one user.

Report Agent: global counters

Counter Name	Object Name	Instance Name	Description
IsReporting	TCReport_Report _General	TCREPORT_Rep ort	Is “1” during report creation, else “0”. Can be seen in KCS Monitor with info “creating”.

Scripts for Manual Database Backup and Cleanup

The Fetch Agent setup installs batch jobs that can be used for manual backup or deletion of old information. These scripts are currently only supported for a local database (on the same machine as TCReport Fetch Agent).

Important Restrictions:

For these batch jobs, the SQL user account of the TC/Report Fetch Agent needs the following permissions:

EXECUTE permission for the master database stored procedure sp_bcp_dbcmptlevel.

CREATE DATABASE permission in the master database (for TIMED_BACKUP and for the optional backup database created by TIMED_CLEANUP).

If these permissions are not granted, the batch jobs will fail.

For the batch jobs to work correctly, the following values must consist of characters from the local Windows code page only:

- database name
- SQL server instance name
- SQL server user name
- SQL server user password
- folder used for temporary files
- backup database name

TIMED_BACKUP.BAT

This batch job copies old information (all before a specified date) from the database to a backup database. Copying is done via batch export / import using temporary files.

Parameters:

- Name of backup database (will be created in the same directory as the TCReport database).
- Minimum date for which information shall not be copied (yyyy-mm-dd)
- Password of TCReport database user
- Directory for temp files (information exported from database)

Example:

TIMED_BACKUP TCReport_Last_Year 2003-01-01 password D:\TEMP

A backup database named TCReport_Last_Year is created.
This database is filled with the following information:

- all records from Server_Table and Channel_Table
- all records with time stamp prior to 2003-01-01 from Action_Table and Log_Table

- all User_Table records that were created before 2003-01-01
- if Probe Agent installed on the same computer: all Probe_Table records with a start date before 2003-01-01 (TCOSS time)
- if SLA Monitoring is installed: all records from the AppDownTime_Table with DownFrom date before 2003-01-01

Note Other TC/Report and TC/Probe tables are created in the backup database, but not filled with data.

TIMED_CLEANUP.BAT

This batch job performs a cleanup of the TCReport database: old information (everything before a specified date) is removed from the database.

Parameters:

- Start date for information that remains in the database.
- Password of TC/Report database user.
- Directory for temp files (information exported from database)
- Optional: name of backup database

Example:

TIMED_CLEANUP 2003-01-01 password D:\TEMP BKUPDB

In this example a backup database is specified, and therefore a database BKUPDB is created. The following information is stored there:

- all records from Server_Table and Channel_Table
- all records from Action_Table and Log_Table
- all User_Table records
- if Probe Agent installed on the same computer: all Probe_Table records

Subsequently, the following information is exported to temporary files:

- all records with time stamp greater than 2003-01-01 from Action_Table and Log_Table
- all User_Table records valid on 2003-01-01 or later
- if Probe Agent installed on the same computer: all Probe_Table records with start date after 2003-01-01 (TCOSS time)

Then, the tables mentioned above are truncated, and the information is imported back from the temporary files. Thus, all older information is removed from the database. As a side effect, the identity fields (ID_Action, ID_Log, ID_User) are reset and an overrun of these fields is prevented.

Note This command changes the TC/Report database. Therefore, all processes accessing the database should be stopped while the batch job is running (especially Fetch Agent and Report Agent).

Combining TIMED_BACKUP and TIMED_CLEANUP

You can combine both batch jobs: First run TIMED_BACKUP to save old information to a backup database. Then run TIMED_CLEANUP to delete the old information from the TCReport database.

Notes:

Setup inserts information from the TC/Report configuration into the batch files (database name, server name, TC/Report user name).

The batch files use the SQL user account created for the TC/Report Fetch Agent.

New databases created by TIMED_BACKUP are placed into the folder defined in the Fetch Agent registry value General\DBDirectory. As a rule, if you copy the complete database via TIMED_BACKUP, the required disk space is <size of the existing database> + 100 MB.

The temporary files created during backup and cleanup may be rather big (50 % to 100 % of the database size). Please use a temp directory that has enough free space.

The batch files TIMED_BACKUP.BAT and TIMED_CLEANUP.BAT must not be used in conjunction with the general backup functionality of the FETCHAGENT (configured in the TC Management Console).

Database Upgrade to Unicode

This section explains how you can create a Unicode-enabled copy of a TC/Report database that was created and filled with data by a fetch agent of KCS versions below 9.2.

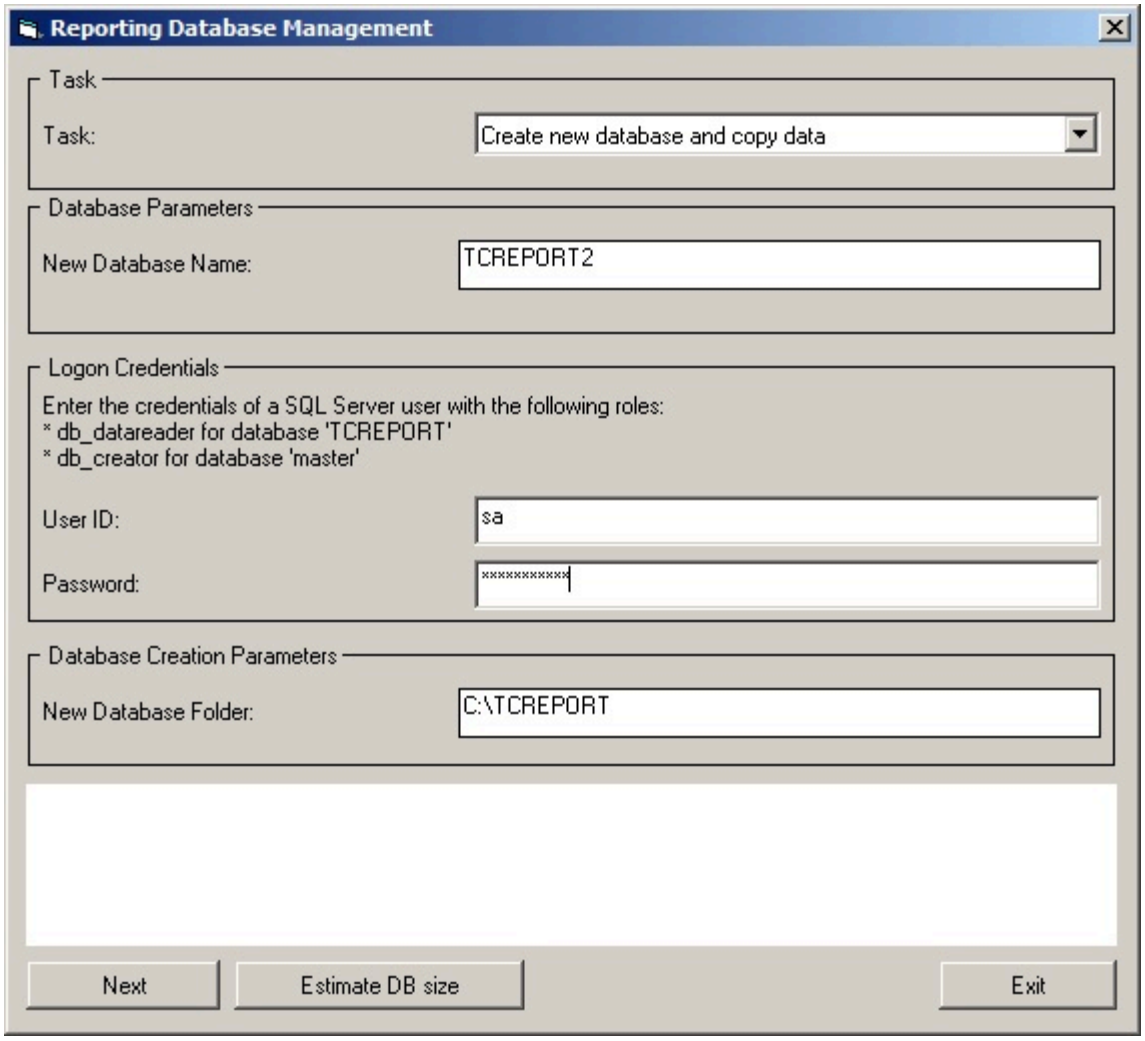
Space requirements for the new database:

The new database needs about 31 percent more space than the data file of the existing database.

Steps:

1. Upgrade the Fetch Agent via KCS Setup version 9.2 or higher, using database creation option “UPDATE DATABASE”. If TCProbe is installed on the same database, upgrade also TCPROBE with the same KCS Setup, using option “UPDATE DATABASE”.
2. Upgrade the TC/Report Report Agent, using the same KCS Setup version.
3. Optionally: Create a new database manually, as described in section Database and User Created by Customer.
4. Stop all programs that access the existing TC/Report database.
5. Log on to the Fetch Agent computer as an administrator, open the command prompt and change to folder C:\TOPCALL\SHARED.
6. Start the following command:
   TCREPSETUP TCREPORT_Fetch -i
   (Character i stands for Interactive)

The program displays the following screen:



The Task combo box offers two options:

- Create new database and copy data

  For this option, you need a SQL Server user with **db_creator** permissions. This user must also have read permissions for the existing TC/Report database.
- Copy data into prepared database

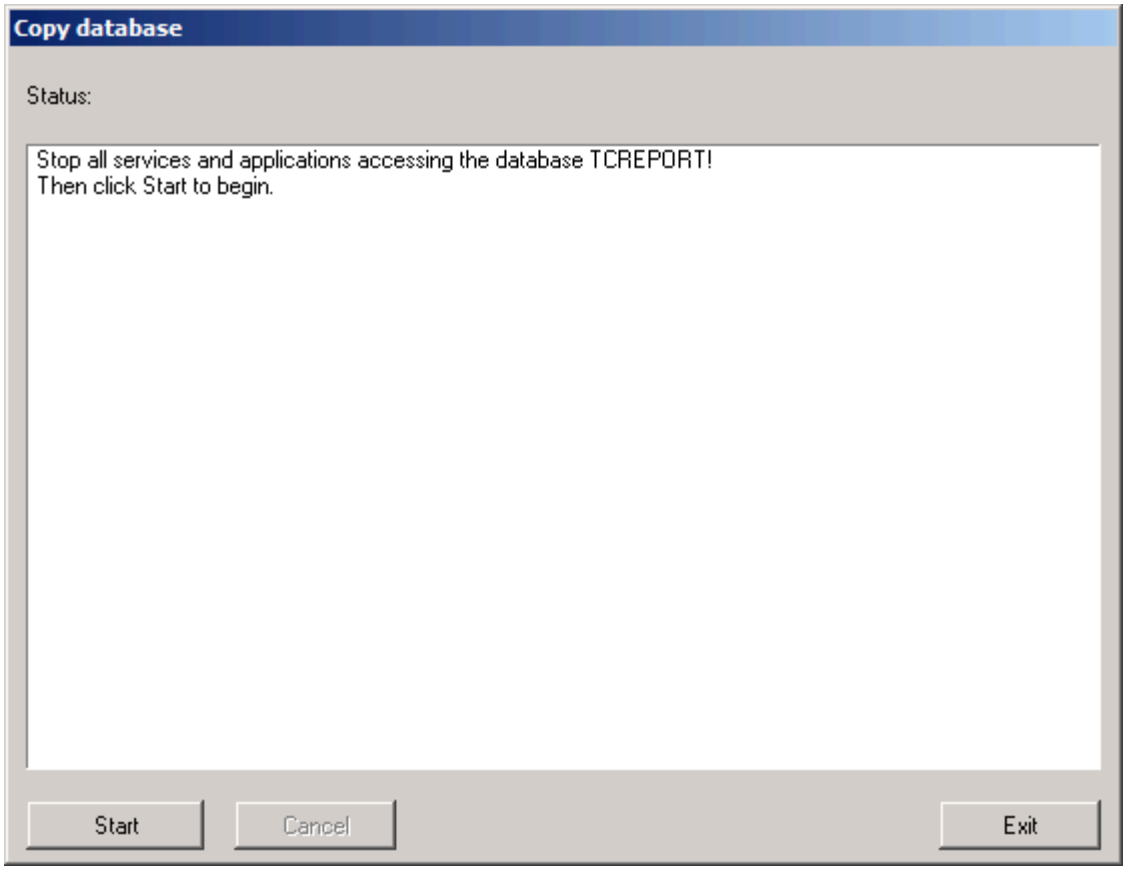
  For this option, you need an existing database and a SQL Server user with **db_owner** permissions for this database. This user must also have read permissions for the existing TC/Report database.

Note
The logon credentials entered here are used only for creating and filling the new database.
The TCReport agents will still use the same SQL Server login as with the existing database.

In both modes, you have to enter the name of the new database. With “create new database and copy data”, you also must specify the folder (on the SQL server) where the database shall be created.

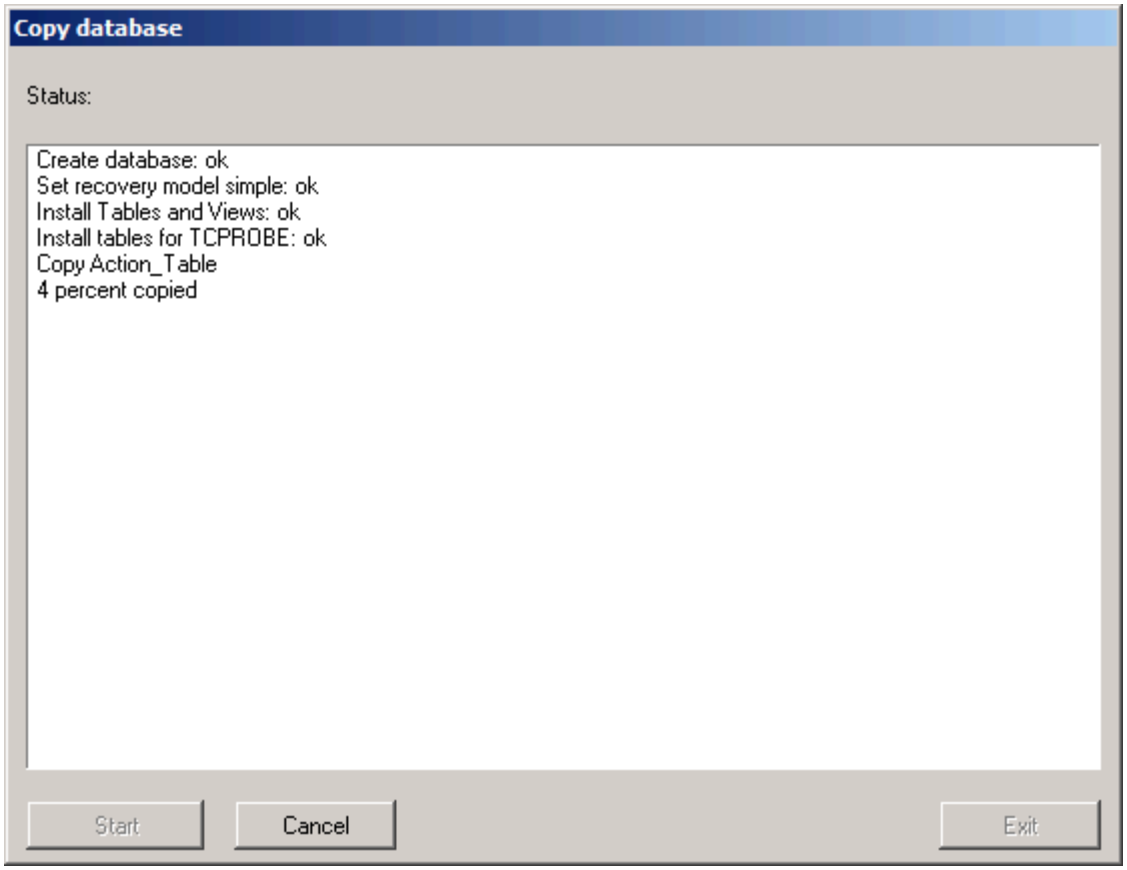
If you click the button **Estimate DB size**, the size of the migrated database is calculated and displayed beside the button.

Click Next to continue.



Click Start to start the database migration.

The major steps of the migration are shown in the window.



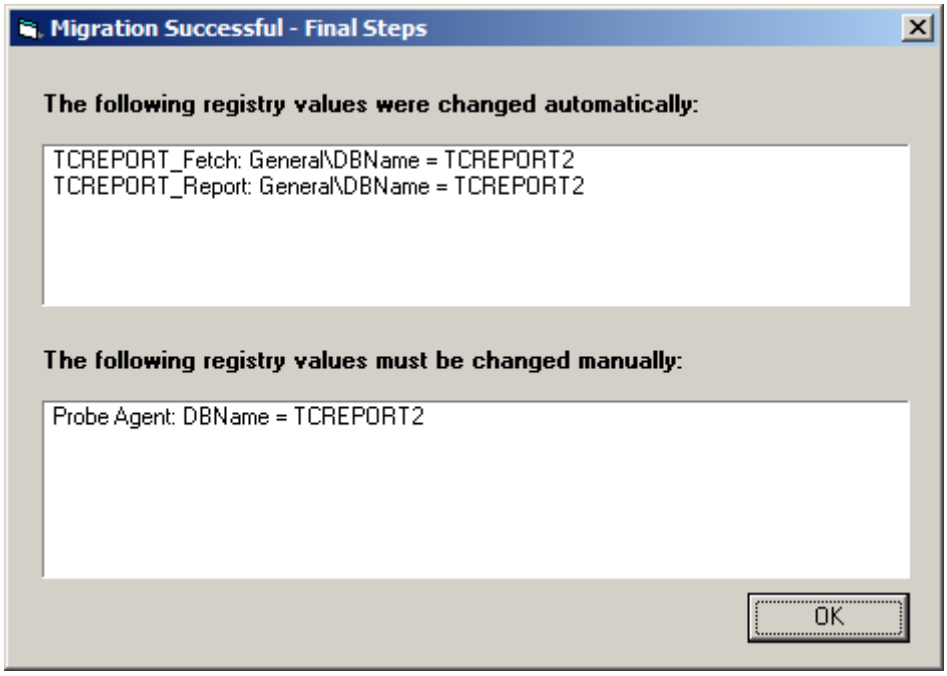
For large databases, the **Action_Table** related tasks can take quite some time.

The database migration can be aborted by clicking **Cancel** (the button is only enabled while the migration is in progress).

If the migration runs into an error or is cancelled, all changes are automatically undone.

The batch jobs timed_backup and timed_cleanup are adjusted automatically to use the new database name.

The name of the new database is also written to the registry of any locally installed TCREPORT or TCPROBE agents that used the old database. If the Report Agent or TCPROBE are running on another computer, you have to adjust their registry manually. The following dialog box shows which changes were done during migration and which changes are still to do manually.



After changing those registry values that were not changed automatically, restart the TC/Report agents.

Note Backup databases are not migrated, - they can still be used for report generation. Nevertheless, any backups created after the migration will use new database names.

Upgrade TC/Report Database

If you are upgrading Kofax Communication Server from version 10.2.0 or lower, do the following to upgrade the TC/Report database.

Note

- This is not required for fresh installations of Kofax Communication Server.
- You must upgrade the TC/Report database before upgrading the Kofax Communication Server setup.

1. Rename the existing Log_Table to Log_Table_Backup.
2. To create a new Log_Table, run the following script in SQL Analyzer:

```
USE [TCReport]
GO

/****** Object:  Table [dbo].[Log_Table]     ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
```

```
GO

CREATE TABLE [dbo].[Log_Table](
 [ID] [bigint] IDENTITY(1,1) NOT NULL,
 [Server_ID] [nvarchar](30) NOT NULL,
 [Time_Stamp_Local] [datetime] NOT NULL,
 [LOG_ID] [varchar](128) NOT NULL,
 [Object_Name] [nvarchar](128) NOT NULL,
 [Value_Type] [int] NOT NULL,
 [Value_String] [nvarchar](128) NULL,
 [Value_Int] [int] NULL,
 [Time_Stamp_UTC] [datetime] NOT NULL,
 [Msg_ID] [bigint] NOT NULL,
 [First_ID] [bigint] NULL,
 [Log_User] [nvarchar](128) NULL,
 CONSTRAINT [PK__ID9] PRIMARY KEY CLUSTERED
(
 [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
 ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO
```

3. To copy data from Log_Table_Backup to the Log_Table, run the following script in SQL Analyzer:

```
SET Identity_insert [TCReport].[dbo].[Log_Table] ON
INSERT INTO   [TCReport].[dbo].[Log_Table]
(
 [ID] ,
 [Server_ID] ,
 [Time_Stamp_Local],
 [LOG_ID],
 [Object_Name],
 [Value_Type],
 [Value_String],
 [Value_Int],
 [Time_Stamp_UTC],
 [Msg_ID],
 [First_ID],
 [Log_User]
)
SELECT * FROM  [TCReport].[dbo].[Log_Table_Backup]
SET Identity_insert [TCReport].[dbo].[Log_Table] off
```

4. To create indexes in the Log_Table, run the following script in SQL Analyzer:

```
USE [TCReport]
GO

/****** Object:  Index [IX_Server_ID]     ******/
CREATE NONCLUSTERED INDEX [IX_Server_ID] ON [dbo].[Log_Table]
(
 [Server_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
 DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
 ON [PRIMARY]
GO


/****** Object:  Index [IX_Time_Stamp_Local]   ******/
CREATE NONCLUSTERED INDEX [IX_Time_Stamp_Local] ON [dbo].[Log_Table]
```

```
(
 [Time_Stamp_Local] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
 DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
 ON [PRIMARY]
GO
/****** Object:  Index [IX_Time_Stamp_UTC]    ******/
CREATE NONCLUSTERED INDEX [IX_Time_Stamp_UTC] ON [dbo].[Log_Table]
(
 [Time_Stamp_UTC] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
 DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
 ON [PRIMARY]
GO


/****** Object:  Index [IX_Log_ID]    ******/
CREATE NONCLUSTERED INDEX [IX_Log_ID] ON [dbo].[Log_Table]
(
 [LOG_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
 DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
 ON [PRIMARY]
GO


/****** Object:  Index [IX_Object_Name]   ******/
CREATE NONCLUSTERED INDEX [IX_Object_Name] ON [dbo].[Log_Table]
(
 [Object_Name] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
 DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
 ON [PRIMARY]
GO


/****** Object:  Index [IX_Msg_ID]   ******/
CREATE NONCLUSTERED INDEX [IX_Msg_ID] ON [dbo].[Log_Table]
(
 [Msg_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
 DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
 ON [PRIMARY]
GO
```

Fetch Agent Registry

Path: HKEY_LOCAL_MACHINE\SOFTWARE\Topcall\TCReport_Fetch

Entry	Type	Default Value	Description
CommandLine	REG_SZ	C:\TOPCALL\Shared\TCFetch.exe	Path to the Fetch agent application
LogonType	REG_SZ	Batch	
MaxTraceFiles	REG_DWORD	2	Maximal number of trace files
MaxTraceFileSize	REG_DWORD	0x3e8	Maximal size of trace files
Password	REG_SZ		Windows Password

Entry	Type	Default Value	Description
TraceLevel	REG_DWORD	1	1 = Standard 0xff (255) = Debug
UserId	REG_SZ		Windows User
Domain	REG_SZ		Windows Domain

Path: HKEY_LOCAL_MACHINE\SOFTWARE\Topcall\TCReport_Fetch\General

Entry	Type	Default Value	Description
DailyFullFetch	REG_SZ	“”	Time for daily full fetch of user data (hhmm)
DBBackupBatch	REG_SZ	“”	the batch file (with path)
DBBackupPercent	REG_DWORD	0x64 (100 %)	Tuning value for backup size limit estimation
DBBackupPeriod	REG_DWORD	0	0=day, 1=week, 2=month, 3=quarter, 4=half year, 5=year
DBBackupPeriodFactor	REG_DWORD	0	e.g. factor 2 means 2 month, or 2 days, …
DBBackupPeriodic	REG_DWORD	0	checkbox: backup is periodic
DBBackupSendMsg	REG_DWORD	0	checkbox: TC/Report shall send a message
DBBackupSendUser	REG_SZ	“”	Recipient
DBBackupSizeIsLimited	REG_DWORD	0	checkbox: if there is a size limit
DBBackupSizeLimit	REG_DWORD	0	the size limit
DBBackupStartBatch	REG_DWORD	0	checkbox: start a batch file
DBCollation	REG_SZ		Collation
DBDirectory	REG_SZ	C:\MSSQL7\DATA	Directory for database (on SQL server)
DBMaxBackup	REG_DWORD	0	maximum number of backup files
DBMaxDataSize	REG_DWORD	0	Maximum size of database content (MB). If not 0, this value overrides the DBMaxSize setting.
DBMaxSize	REG_DWORD	1900	Maximal size of database file (MB). 0 disables size check.
DBName	REG_SZ	TCREPORT	Database name
DBServerName	REG_SZ	“”	Name of the SQL server
DBServerType	REG_SZ	MS	Reserved for future use
DBSetupError	REG_DWORD	0	Used by Setup only
DBSetupErrorTxt	REG_SZ	“”	Used by Setup only

Entry	Type	Default Value	Description
DBShrinkAsync	REG_DWORD	1	Affects deletion of oldest entries 1: deleting old entries and fetching new entries alternates 0: no new entries are fetched until enough old entries are deleted and database size is okay again
DBSystemPassword	REG_SZ	“”	System user password (for Setup only)
DBSystemUser	REG_SZ	“sa”	System user name (for Setup only)
DBUserName	REG_SZ	“”	Database user used for fetching
DBUserPassword	REG_SZ	“password”	Encrypted password of database user used for fetching
WithAppPanel	REG_DWORD	0	1 if application downtime fetching enabled
MaxDays	REG_DWORD	3	Specifies how many days of event log information are fetched for a new workstation. If 0, everything is fetched.
FetchDelay	REG_DWORD	20	delay during the fetch
FetchPollCycle	REG_DWORD	0x3c	Poll cycle for fetch agent in seconds
WithActionTable	REG_DWORD	1	1: write information to the Action_Table (old operation mode, needed for Crystal Reports based reports) 0: do not write information to the Action_Table (saves disk space, if only TC/MA is used for reporting)
ShrinkLog	REG_DWORD	1	1: install standard SHRINK_LOG procedure (for automatic truncation of transaction log) 0: install dummy SHRINK_LOG procedure (truncation of transaction log is up to the administrator)

Entry	Type	Default Value	Description
SkipUserFetch	REG_DWORD	0	Use ONLY after upgrade from TC/Report versions below 2.10.00, and ONLY if no user changes have been done during the upgrade. The purpose of this value is to make start up after the version upgrade faster, by disabling the time-consuming user fetch. A value of 1 disables full user fetch at first start up after the upgrade (when Last_Change_US and Last_Change_RS in the server table are NULL). This registry value must be created manually. This registry key is set to value 0 after one poll cycle.
DBProvider	REG_SZ	MSOLEDBSQL	To support TLS1.2
MultiSubnetFailOver	REG_DWORD	1	To support SQL High Availability Cluster.
ConnectionTimeout	REG_DWORD	15	To support SQL High Availability Cluster, set the value as '90'. If required, you can increase this value.

Path: HKEY_LOCAL_MACHINE\SOFTWARE\Topcall\TCReport_Fetch\TCMA

The following registry values are for integration with TC/MA:

Entry	Type	Default value	Description
WithEvents	REG_DWORD	0	0: do not fetch message tracking information (old operation mode) 1: fetch message tracking information needed for OmniAnalyser (for all servers where it is not explicitly disabled)
DBName	REG_SZ	“EVENTS”	Only applicable if WithEvents = 1. Name of the events database. If equal to TCREPORT database, DBPath is ignored.
DBPath	REG_SZ	Same as General \DBDirectory	Only applicable if WithEvents = 1. Folder for the events database.
DBCleanupTime	REG_SZ	00:00	Time (local time) for event database cleanup
DBCleanupDays	REG_DWORD	30	Number of days that information must be kept

Entry	Type	Default value	Description
DBLastCleanup	REG_SZ	2000-01-01	Date (local time) of last event database cleanup
PrefixUserNames	REG_DWORD	0	If 1, user names are prefixed with server name
UseReceiveQueue	REG_DWORD	1	If 1: all messages posted to TCOSS via TCLINK create a Transfer In event, media type preserved If 0: Messages from shadow users posted via TCLINK create a Submit event, media type is lost For developer only.
TCDomain	REG_SZ	TOPCALL	The fictitious root domain name used for KCS media types (e.g. fax.topcall, sms.topcall)

FetchWorkStations Registry

Path: HKEY_LOCAL_MACHINE\SOFTWARE\Topcall\FetchWorkStations

Entry	Type	Default Value	Description
CommandLine	REG_SZ	C:\TOPCALL\Shared \FetchWorkStations.exe	Path to the FetchWorkStations executable
LogonType	REG_SZ	Batch	
MaxTraceFiles	REG_DWORD	2	Maximal number of trace files
MaxTraceFileSize	REG_DWORD	0x3e8	Maximal size of trace files
Password	REG_SZ		Windows Password
TraceLevel	REG_DWORD	1	0 errors only >= 1 Standard (minimal) >= 10 SQL and WQL statements >= 20 function calls
PollCycleSec	REG_DWORD	300	Poll cycle (wait time between activity phases)
WmiTimeoutSec	REG_DWORD	60	Timeout for WMI actions (connect to remote computer, retrieve next event from remote computer)
Retry	REG_DWORD	0	This setting controls retry behavior when a workstation could not be reached via WMI. 0: retry next day 1: retry immediately
UserId	REG_SZ		Windows User

Entry	Type	Default Value	Description
Domain	REG_SZ		Windows Domain

FetchWorkStations.exe reads the database connection parameters from the registry of the locally installed TCREPORT_Fetch.

Report Agent Registry

Path: HKEY_LOCAL_MACHINE\SOFTWARE\Topcall\TCReport_Report

Entry	Type	Default Value	Description
CommandLine	REG_SZ	C:\TOPCALL\Shared\ TCReport.exe	Path to the Report agent application
MaxTraceFiles	REG_DWORD	2	Maximal number of trace files
MaxTraceFileSize	REG_DWORD	0x3e8	Maximal size of trace files
TraceLevel	REG_DWORD	1	1 = Standard 0xff (255) = Debug
Password	REG_SZ		
UserId	REG_SZ		Windows User
LogonType	REG_SZ	Batch	
Domain	REG_SZ		Windows Domain

Path: HKEY_LOCAL_MACHINE\SOFTWARE\Topcall\TCReport_Report\General

Entry	Type	Default Value	Description
DBName	REG_SZ	TCREPORT	Database name
DBServerName	REG_SZ		SQL server name
DBServerType	REG_SZ	MS	Reserved for future use
DBUserName	REG_SZ	“”	Database user used for connecting Report agent
DBUserPassword	REG_SZ	“”	Encrypted password of database user
DBPausedStateCheck	DWORD	0	Check the ‘paused’ state for SQL DB. The possible values are '0','1' and '2'. 0: Before each report creation, TCReport checks for the ‘paused’ state by sending a login request with a wrong password. 1: The check for paused state is skipped. 2: Resets the wrong login counter to ‘0’. Tries to establish a connection with correct credentials.

Entry	Type	Default Value	Description
ExcelAreaGroupNumber	DWORD	ffffffff (-1)	For Excel reports: If column width is based on a group header or footer (ExcelAreaType = 3 or 5), this value defines which group (1,2,…) is meant. -1 means: not specified Registry value created at Report Agent start.
ExcelAreaType	DWORD	8	For Excel reports: Defines on which report area the column width is based. Possible values: 1: Report header 2: Page header 3: Group header (group number according to ExcelAreaGroupNumber) 4: Details 5: Group footer (group number according to ExcelAreaGroupNumber) 7: Page footer 8: Report footer 255: Whole report Value 8 is good for all standard reports delivered with the product. Other values may be needed for custom reports (designed by technician or customer), if their layout is different than the standard report layout. Registry value created at Report Agent start.
ExcelVersion	DWORD	27	Defines the Excel version for Excel reports: Possible values (decimal): 21: Excel50 22: Excel50 Tabular 27: Excel70 28: Excel70 Tabular 29: Excel80 30: Excel80 Tabular 36: Excel97 Registry value created at Report Agent start.
HtmlEnableSeparatedPages	REG_DWORD	1	0: one HTML file created 1: one HTML file per page created Registry value created at Report Agent start.
HtmlHasPageNavigator	REG_DWORD	1	0: HTML report has no page navigator 1: HTML report has a page navigator on each page. Registry value created at Report Agent start.

Entry	Type	Default Value	Description
HtmlVersion	REG_DWORD	32	Defines the HTML version of HTML reports. Possible values (decimal): 32: HTML 4.0 24: HTML 3.2 Standard 25: HTML 3.2. Extended Registry value created at Report Agent start.
LngFilePath	REG_SZ	C:\TOPCALL \Shared \TCRPT01	Path to the language used during creating reports
Notif	REG_SZ	1h;24h	Default timing for NOTIF report
PrintTimeout	REG_DWORD	0x1e	Maximum wait time (sec) for printing to KCS printer, for Destination=FAX
ReportPath	REG_SZ	C:\TOPCALL \Shared\RPT	Path to the RPT files
ReportPollCycle	REG_DWORD	0x3c	Poll cycle for report agent in seconds
RuntimeVersion	REG_SZ	11	Version of the Crystal Reports Runtime library. Do not change.
ReportVersion	REG_SZ	10	Used by Setup only. Do not change.
StdReports	REG_SZ	TC	Prefix for new standard reports
SendInfoMessage121	DWORD	1	Set this value to 0 if the message “The report will be created later” shall be suppressed.
CsvStringSeparator	REG_SZ	“	String separator for CSV format
DBProvider	REG_SZ	MSOLEDBSQL	To support TLS1.2
MultiSubnetFailOver	REG_DWORD	1	To support SQL High Availability Cluster.
ConnectionTimeout	REG_DWORD	15	To support SQL High Availability Cluster, set the value as '90'. If required, you can increase this value.

Path: HKEY_LOCAL_MACHINE\SOFTWARE\Topcall\TCReport_Report\Topcall

Entry	Type	Default Value	Description
ServerPath	REG_SZ	“”	Path to TCOSS for Report Agent (e.g. TCP/IP,DEMOTC)
UserID	REG_SZ	“”	User ID – used for connection to KCS
UserPassword	REG_SZ	“”	User password – used for connection to KCS, encrypted
FISPrefix	REG_SZ	REP	Prefix for INI files in FIS folder. If empty: no INI files created
MaxRequest Count	REG_DWORD	1000	Number of report requests before agent is restarted automatically

SSL/TLS Registry Keys

To support TLS 1.2 security, add the following registry keys and set the values as described:

SSL 2.0

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL
\Protocols\SSL 2.0\Client
```

Name	Type	Value
DisabledByDefault	REG_DWORD	1
Enabled	REG_DWORD	0

SSL 3.0

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL
\Protocols\SSL 2.0\Server
```

Name	Type	Value
DisabledByDefault	REG_DWORD	1
Enabled	REG_DWORD	0

TLS 1.0

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL
\Protocols\TLS 1.0\Client
```

Name	Type	Value
DisabledByDefault	REG_DWORD	1
Enabled	REG_DWORD	0

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL
\Protocols\TLS 1.0\Server
```

Name	Type	Value
DisabledByDefault	REG_DWORD	1
Enabled	REG_DWORD	0

TLS 1.1

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL
\Protocols\TLS 1.1\Client
```

Name	Type	Value
DisabledByDefault	REG_DWORD	1
Enabled	REG_DWORD	0

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.1\Server`

Name	Type	Value
DisabledByDefault	REG_DWORD	1
Enabled	REG_DWORD	0

TLS 1.2

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Client`

Name	Type	Value
DisabledByDefault	REG_DWORD	0
Enabled	REG_DWORD	1

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server`

Name	Type	Value
DisabledByDefault	REG_DWORD	0
Enabled	REG_DWORD	1

Accessing the Database

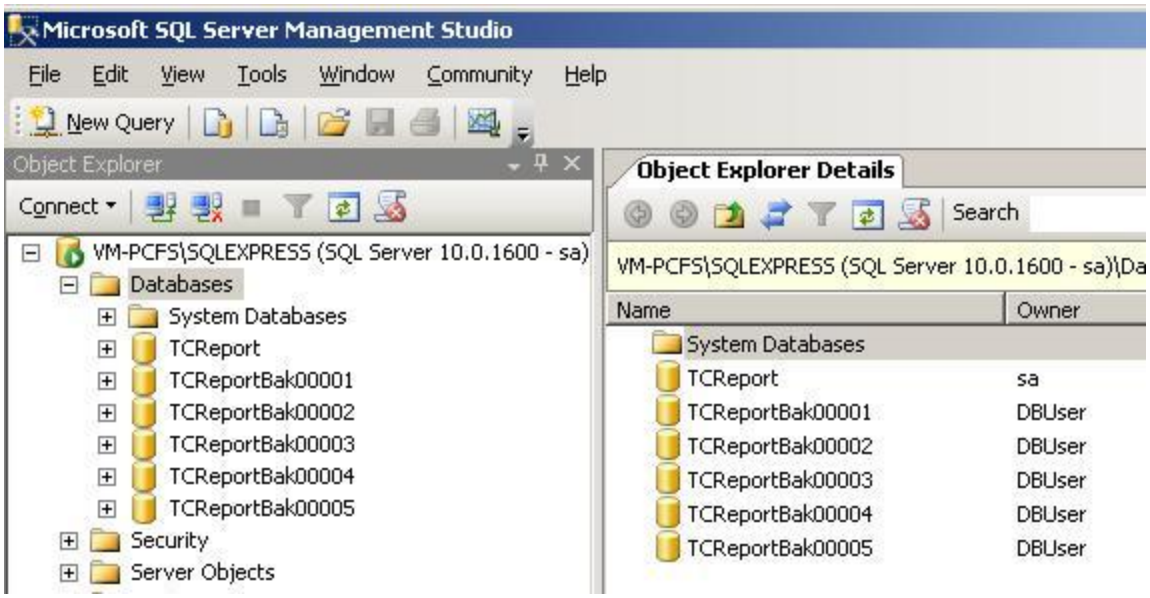
If you install “Microsoft SQL Server Express with Tools”, you can use “SQL Server Management Studio” to access the TC/Report database.

For SQLExpress 2005, a standalone download named “Microsoft SQL Server Management Studio Express” is available:
http://www.microsoft.com/downloadS/details.aspx?familyid=C243A5AE-4BD1-4E3D-94B8-5A0F62BF7796&displaylang=en

Microsoft SQL Server 2008 Express with Tools can be downloaded from:

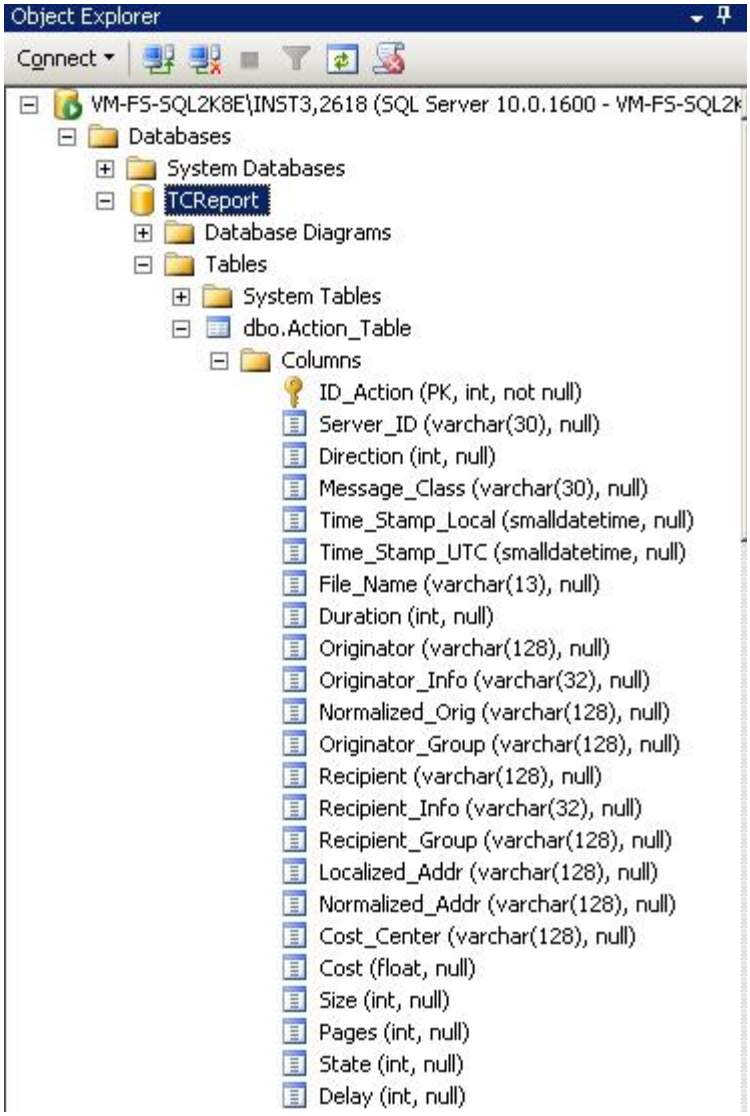
http://www.microsoft.com/downloads/details.aspx?FamilyId=7522A683-4CB2-454E-B908-E805E9BD4E28&displaylang=en

SQL Server Management Studio

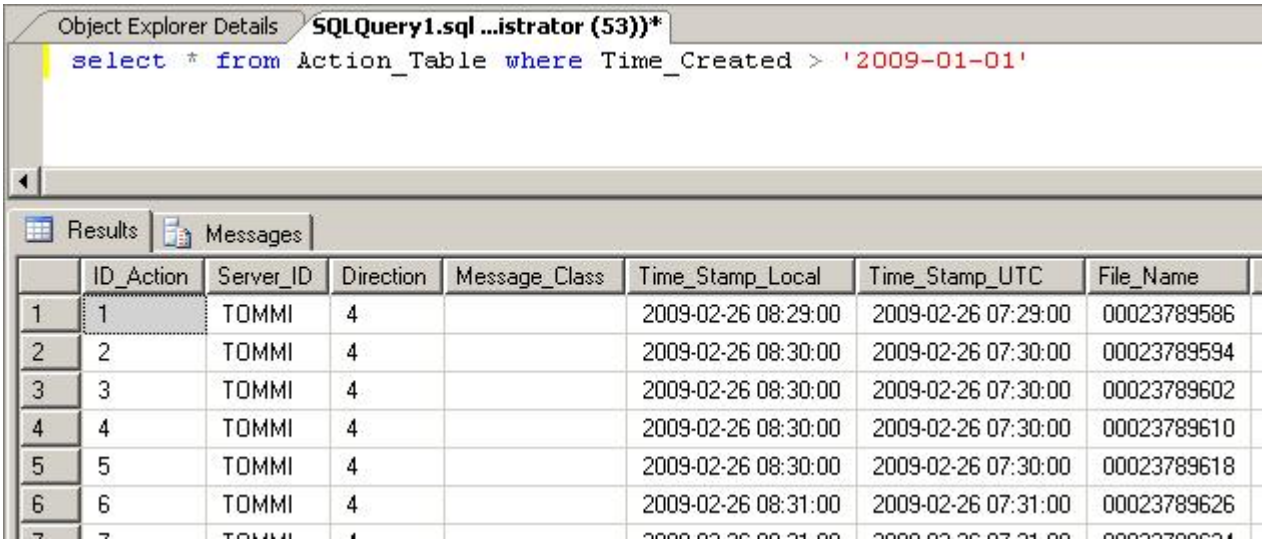


The above screen shot shows where the TCReport database (and optional backup databases) can be found in SQL Server 2008 Management Studio.

You can view the structure of all objects in the TCReport database, e.g. the columns of the Action_Table:



The tool includes an SQL query editor. The following screen shot shows how to search for Action_Table entries with a certain creation time:



	ID_Action	Server_ID	Direction	Message_Class	Time_Stamp_Local	Time_Stamp_UTC	File_Name
1	1	TOMMI	4		2009-02-26 08:29:00	2009-02-26 07:29:00	00023789586
2	2	TOMMI	4		2009-02-26 08:30:00	2009-02-26 07:30:00	00023789594
3	3	TOMMI	4		2009-02-26 08:30:00	2009-02-26 07:30:00	00023789602
4	4	TOMMI	4		2009-02-26 08:30:00	2009-02-26 07:30:00	00023789610
5	5	TOMMI	4		2009-02-26 08:30:00	2009-02-26 07:30:00	00023789618
6	6	TOMMI	4		2009-02-26 08:31:00	2009-02-26 07:31:00	00023789626

For more information about possible actions, please consult the online help of this application.

Error Handling

Both agents write severe errors to the application event log and to their trace files. For a complete list of event log entries see the SNMP Integration Manual. This section covers only special error conditions.

TCOSS permission problems

The Fetch Agent does not stop when it encounters a permission problem with one of the configured TCOSS servers. The problem is logged to the event log and displayed in KCS Monitor. The event log entry text explains which rights are missing. It is not necessary to restart the Fetch Agent after granting the missing permissions.

The following TCOSS permission problems can occur:

No permission to list in- and outboxes

If the Fetch Agent cannot list all in- and outboxes, fetching of send orders from this server is not possible. User fetching is not affected. Other TCOSS servers are still fetched.

Error message in KCS Monitor: NO PERMISSION TO LIST ALL IN AND OUTBOXES!

Event logged when the error status starts:

ID	Severity	Text	Parameters
8562	Error	User %1 is not allowed to list all in and outboxes on server %2	%1: user id %2: server id

Event logged when the error status ends, and also after successful startup:

ID	Severity	Text	Parameters
8565	Information	User %1 is allowed to list all in and outboxes on server %2	%1: user id %2: server id

No permission to open in- and outboxes

The Fetch Agent detects this error at the first attempt to fetch a TCOSS log entry. From this moment, fetching of send orders or log entries from this server is not possible. User fetching is not affected. Other TCOSS servers are still fetched.

Error message in KCS Monitor: NO PERMISSION TO OPEN ALL IN AND OUTBOXES !

Event logged when the error status starts:

ID	Severity	Text	Parameters
8563	Error	User %1 is not allowed to open all in and outboxes on server %2	%1: user id %2: server id

Event logged when the error status ends, and also after successful startup:

ID	Severity	Text	Parameters
8566	Information	User %1 is allowed to open all in and outboxes on server %2	%1: user id %2: server id

No permission to read user information

If the Fetch Agent cannot read user information (system user profiles), fetching of user information from this server is not possible. Fetching of send orders and log entries is not affected. Other TCOSS servers are still fetched.

Error message in KCS Monitor: NO PERMISSION TO READ USER PROFILES!

Event logged when the error status starts:

ID	Severity	Text	Parameters
8564	Error	User %1 is not allowed to read user profiles on server %2	%1: user id %2: server id

Event logged when the error status ends:

ID	Severity	Text	Parameters
8567	Information	User %1 is allowed to read user profiles on server %2	%1: user id %2: server id

Error setting “max server memory”

During Setup, the application TCREPSETUP.EXE creates objects on the SQL server (database, tables, views, stored procedures). Setup tries to increase the “max server memory” on the SQL server to 80 MB. Before setting the "max server memory" the Setup program reads the currently configured value.

If this value is already higher than the required minimum value (80 MB), Setup leaves the current value unchanged.

If the configured value is lower than 80MB, the setup program tries to set it to this new value 80 MB.

If this step fails (because setting the max server memory is not allowed), the setup will report this error to the TCREPORT_FETCH trace file and to the event log.

Event:

ID	Severity	Text	Parameters
8556	Warning	'max server memory' cannot be changed: %1 (%2)	%1: explanation %2: error description returned by SQL server

Regardless of the error, Setup continues with the installation.

Error granting “Create Database” permission

During Setup, the application TCREPSETUP.EXE tries to give the SQL user permissions to create new databases. This permission is needed when TC/Report is supposed create backup databases.

If this permission cannot be granted, a warning will be written to the trace file and to the event log. Setup continues with the installation.

Event:

ID	Severity	Text	Parameters
8560	Warning	Setup cannot grant create database to user %1. Error: %2	%1: user id %2: error description returned by SQL server

You can ignore this warning if TC/Report is not configured for backups, if this is an upgrade setup and the backup databases already exist, or if the user already has this permission.
Section Fetch Agent Setup (“Installation” – “Step 3-Setup Agents and Database” – “Fetch Agent Setup”) explains how to run setup without a sysadmin user. In this scenario, the event log warning 8560 is to be expected and can be safely ignored.

Chapter 8

Creating or Modifying Reports

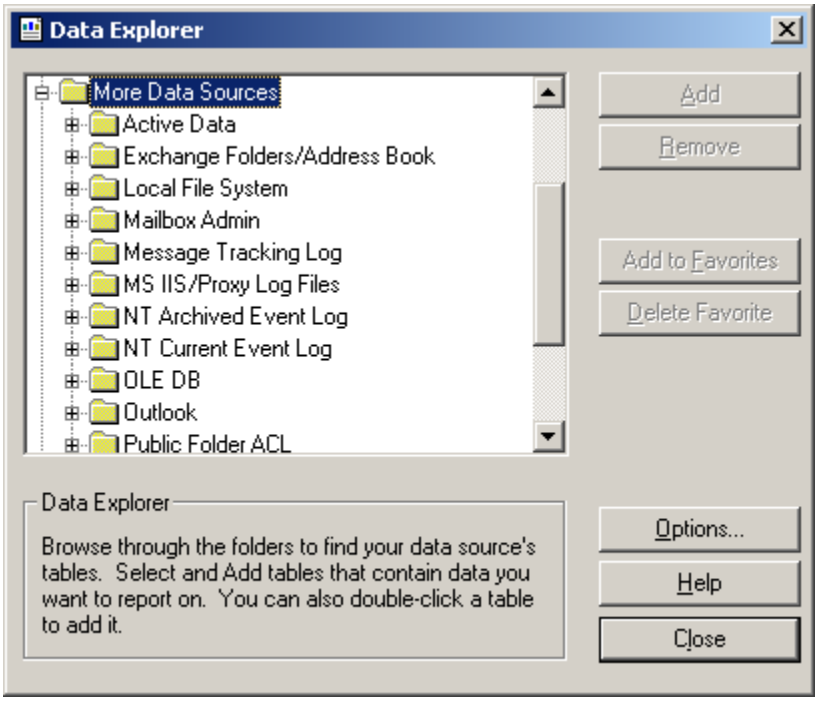
The Report files (*c:\topcall\shared\rpt*.rpt*) can be created and modified with Crystal Reports Designer (version 10, 11 or 2008).

This can be done either by Kofax Development or Kofax Support or by a person with good knowledge of Crystal Reports and SQL. For creation of new configurable reports you should take one of the existing standard reports as an example.

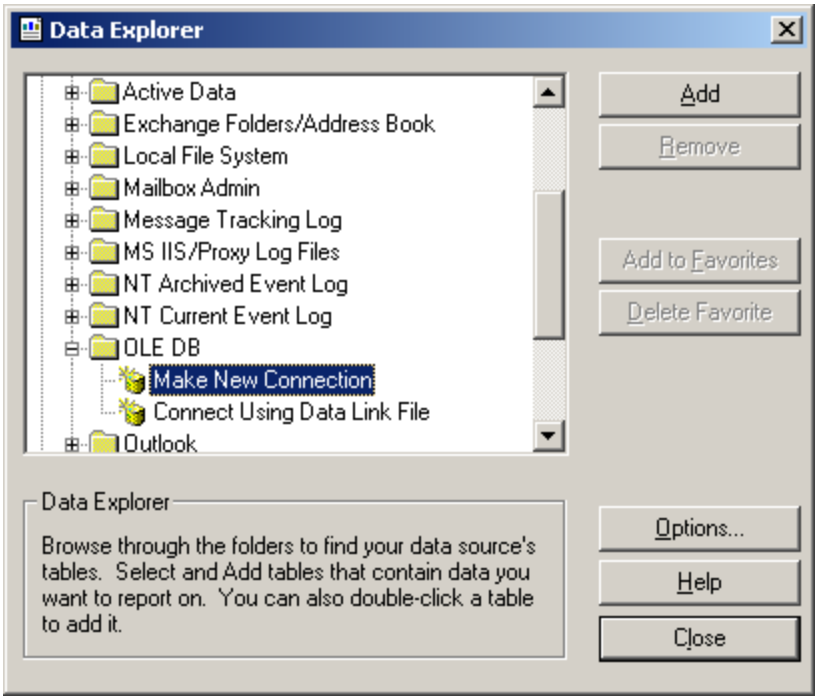
Even for simple modifications, like adding a company logo or changing the scale of graphics, you should be familiar with Crystal Reports. Please consult the help file installed with the Crystal Reports designer for information.

Setting Connection Parameters for New Reports

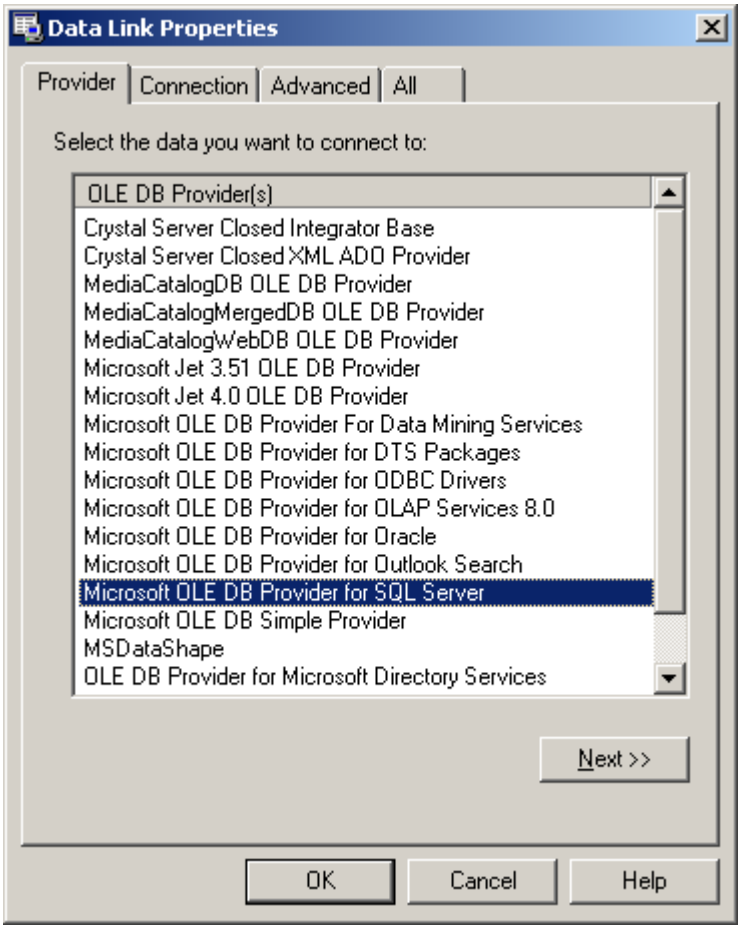
1. TC/Report only works with OLEDB connections. When configuring the database connection in the Data Explorer, expand the node “More Data Sources”.



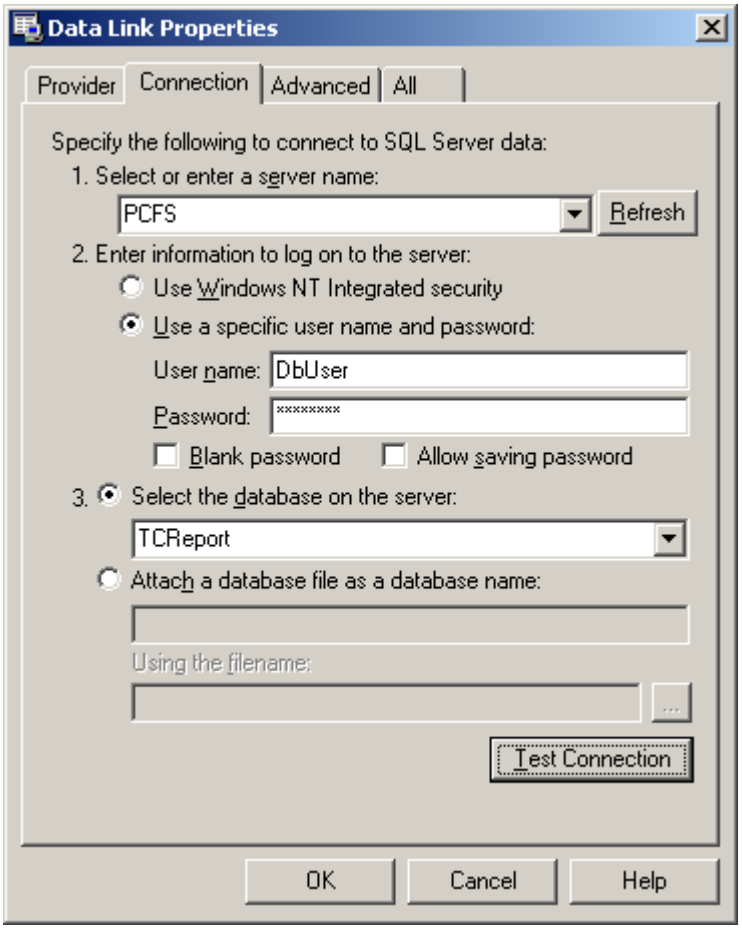
2. Expand the node “OLE DB” and click on “Make New Connection”.



3. Select the provider “Microsoft OLE DB Provider for SQL Server”:



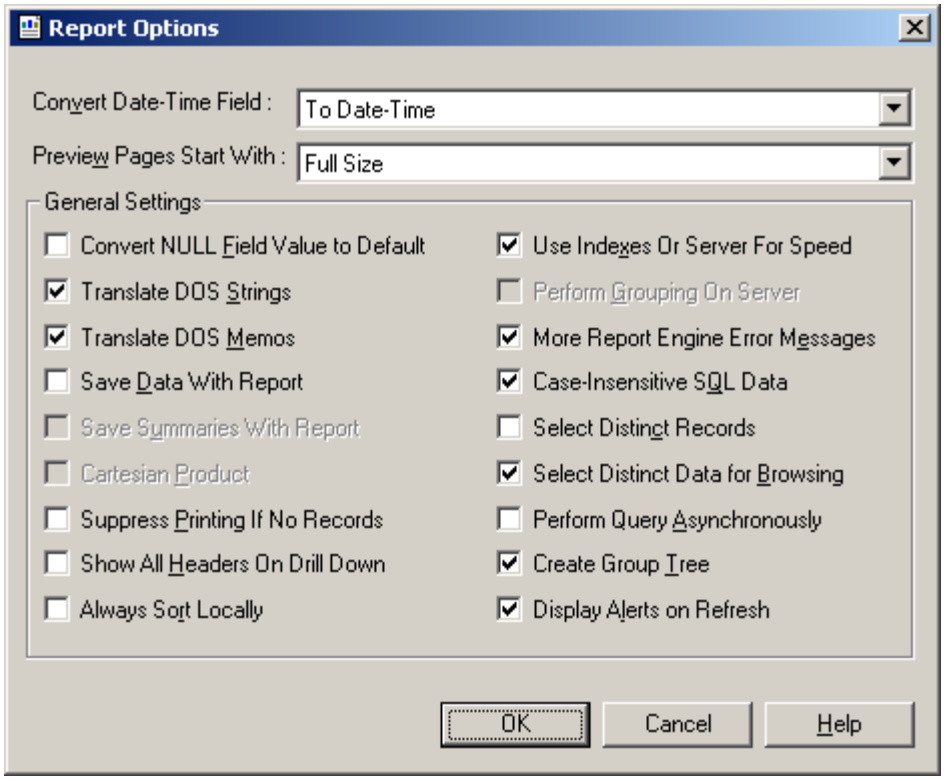
4. Configure access to the database (server name, user credentials, database name).



5. Click **OK**.

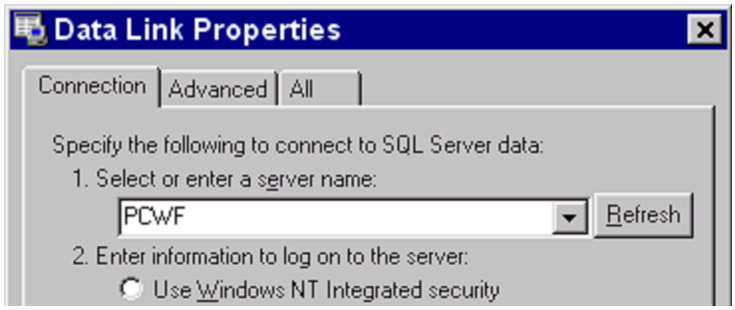
Report Options

In the report options, the checkbox "Save Data With Report" must be cleared!

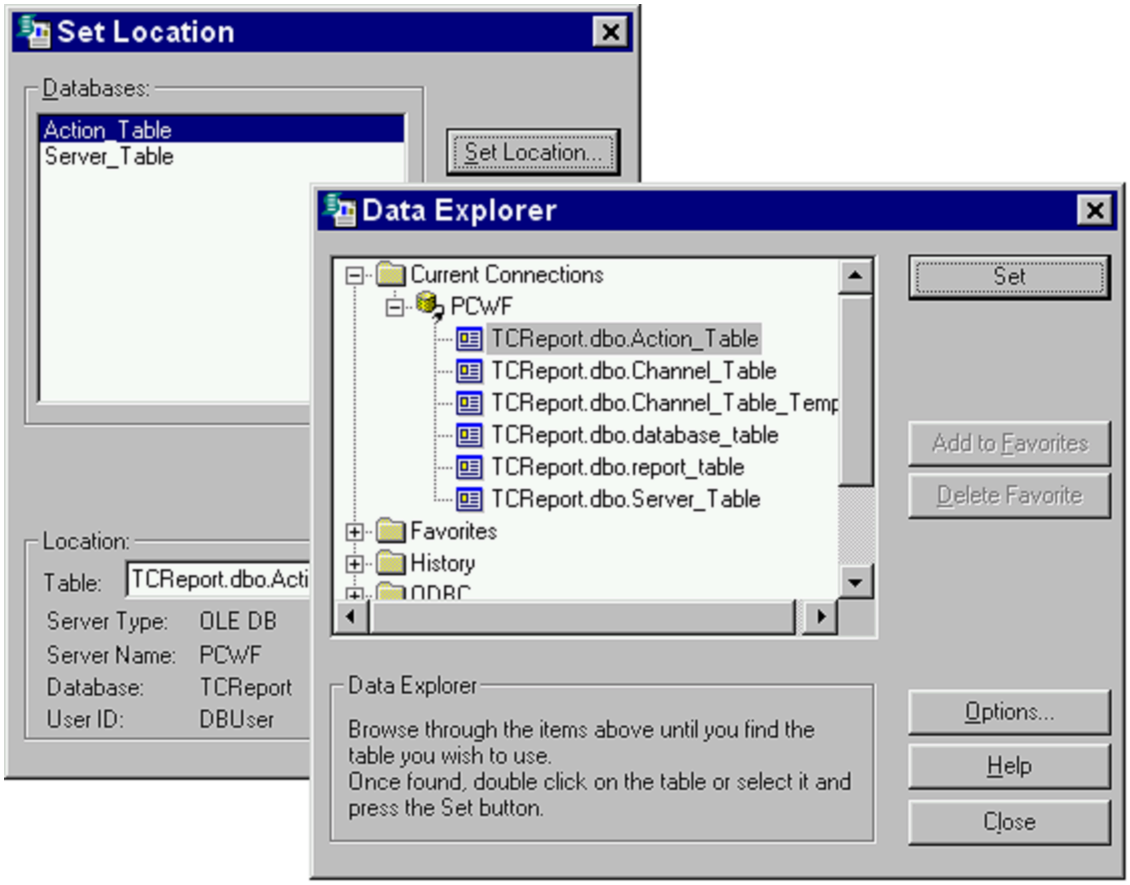


Setting Table Locations When Modifying Reports

If you modify reports, Crystal Reports tries to connect to the database. Enter the computer name of the SQL-Server.



Make sure that the location of every table is set to the current connection (Menu Database -> Set location).



Parameter Fields

The time period is specified by the parameters “Date_From” and “Date_To” (don’t forget the underlines). These parameters hold the TCOSS time frame for the report.

The parameter “Server_ID” specifies the TCOSS server. Default is “*”.

The SQL-Statement of the report should contain the text:

{Action_Table.Time_Stamp_Local} in {?Date_FROM} to {?Date_TO}

Parameters starting with “TEXT_” get the values from the language file *c:\topcall\shared\tcrpt.lng*.

Special Parameters for Integration with TC Management Console

The Report Agent passes special parameters for grouping and result columns to the report types UNIV, DIST, QUEUELEN, TCLINESN, TCJOB, TCDL and LINES. These parameters are needed for integration with the Report Request panel in TC Management Console.

Custom reports must be marked as “standard” reports via a prefix in the report name. The default prefix for custom standard reports is “TC”, this means reports like “TCStatistic” or “TCCost” will receive the parameters mentioned below. The prefix for standard reports can be configured via value *General \StdReports* in the Report Agent registry.

The RPT file name must end with a number that matches the number of groupings used. E.g. the TCLINESN report that has 2 groupings is stored in file TCLINESN2.RPT.

TC/Report handling for standard reports:

- Depending on the number of elements in the report request parameter “GroupOrder”, the Report Agent uses different report files, e.g. “UNIV4.RPT” if 4 groups are requested, UNIV3.RPT for 3 groups etc.
- The elements of the “GroupOrder” parameter are copied to parameters “Group1”, “Group2”, “Group3” and “Group4”.
- Grouping details (list of displayed values) for *GroupX* are written to parameter *ParameterX*.
- The elements of the “ResultOrder” parameter are copied to parameters “Sum1”, “Sum2” and “Sum3”.
- If the Server_Id parameter of the report request contains a list of servers, this list is copied to parameter “Server_List”. The “Server_Id” parameter passed to the Crystal Reports runtime contains only elements with wildcards (e.g. “*” or “Server*”).
- The default orientation for standard reports is Landscape.
- The parameter “Resolution” is modified to contain a valid format string for Crystal Reports.

Example Report Request:

```
ReportName = UNIV
DateFrom = 20010917:000000
DateTo = 20010921:235959
Server_ID = TOM,CHARLIE,NEW*
Wait = NO
GroupOrder = Server_ID,Countrycode_Out,Time
ResultOrder = SendOrders,FailedSendOrders,SuccessfullySent
Resolution = d
internalrecipient = NO
Countrycode_Out = 43
OutputFormat = PDF
```

The Fetch Agent loads UNIV3.RPT and passes the following additional parameters to the Crystal Reports runtime:

```
Group1 = Server_ID
Group2 = Countrycode_Out
Group3 = Time
Parameter1 = *
Parameter2 = 43
Parameter3 = *
Sum1 = SendOrders
Sum2 = FailedSendOrders
Sum3 = SuccessfullySent
Resolution = yyyy/MM/dd
Server_ID = NEW*
Server_List = TOM,CHARLIE
```

INI Files for Standard Reports

In order to integrate a new report with the Report Request wizard, you must provide a matching INI file. Report INI files determine which options are offered to the user and how the user's selections are translated into a textual report request. Copy the file to the RPT directory on the Report Agent workstation. The Report agent then copies it to the TCOSS FIS folder, where it is available for all instances of the Report Request wizard.

Report configuration files are standard Windows INI-Files. All size restrictions for INI files apply. The maximum recommended file size is 32 KB, and the maximum recommended size of a parameter is 32 KB minus 2 bytes.

The INI files are supplied by the author of a report. INI-For the reports delivered with the standard product are part of the TC/Report Setup. Reports without INI-Files can be used, but the Report request wizard cannot create requests for them.

Standard INI files use the local Windows code page.
Unicode INI files with UTF-16 (LE) byte order mark are also supported.

INI File syntax

The following sections must be present in every file:

Section name	Description
General	Report description and version number
Grouping	Information about grouping possibilities
Results	Information about possible statistics results
Parameters	Information about additional parameters supported by this report

There will be additional sections describing individual grouping and result categories and parameters.

Section General

Parameter	Description	Example
Description	Description of this report	Statistic of cost, size and number of messages. Grouped by time period, originator service, recipient service, cost center
Name	Name of this report	UNIVERSAL
Version	Version number	1.0.0

Section Grouping

Parameter	Description	Example
Maximum	Maximum number of grouping categories	4
Minimum	Minimum number of grouping categories	4
ChangeOrder	1 if order can be changed, else 0	1

Parameter	Description	Example
Default	Default grouping categories	GTime,GOriginator,GRecipient,GCostCenter
Names	Available grouping categories	GTime,GOriginator,GRecipient,GCostCenter
SummaryTag	TCReport parameter holding ordered list of categories	GroupOrder

Section *Grouping* tells the wizard which grouping categories are available, whether their order or number may be changed, and which categories are used by default. **This section is needed, even if the grouping cannot be changed**.

Parameters *Default* and *Names* hold a comma-separated list of names. For every name there must be a section with the same name, where the grouping category is described. These are logical names, they do not correspond to the parameter names in the report request. The names must be unique within the INI file.

Parameter *SummaryTag* tells the wizard to write a parameter holding an ordered list of grouping categories into the report request. For new standard reports, the *SummaryTag* must be *GroupOrder*.

Section Results

Parameter	Description	Example
Maximum	Maximum number of result columns	3
Minimum	Minimum number of result columns	3
ChangeOrder	1 if order can be changed, else 0	0
Default	Default result columns	Rcost,Rnumber,Rsize
Names	Available result columns	Rcost,Rnumber,Rsize
SummaryTag	TCReport parameter holding ordered list of result columns	ResultOrder

Section *Results* tells the Wizard, which result columns are available, whether their order or number may be changed, and which result columns are used by default.

This section is needed, even if the number and order of the result columns cannot be changed.

Parameters *Default* and *Names* hold a comma-separated list of names. For every name there must be a section with the same name, where the result column is described. These are logical names, they do not correspond to the parameter names in the report request. The names must be unique within the INI file.

Parameter *SummaryTag* tells the wizard to write a parameter holding an ordered list of result columns into the report request. For new standard reports, the *SummaryTag* value must be *ResultOrder*.

Section Parameters

Parameter	Description	Example
Names	Available additional parameters	PGroup,Pinternal

Section *Parameters* tells the Wizard, which additional parameters are used by this report. The *Names* parameter holds a comma-separated list of names. For every name there must be a section with the same

name, where the result column is described. These are logical names, they do not correspond to the parameter names in the report request. The names must be unique within the INI file.

Sections for individual grouping, column and parameter Names

Parameter	Description	Example
Tag	Parameter name as used in report request text. Use TAG_NONE if there is no corresponding parameter.	Recipient
Description	Describes the type of information	Recipient service
ContentType	One of: CT_FIXED_TIME: represented by position of date and time CT_EDIT_COMBINE: for grouping categories that allow a combination of multiple values CT_FIXED: no separate line, tag will be part of SummaryTag parameter CT_LIST_SINGLE: one of several values CT_EDIT: a value that can be edited	CT_EDIT_COMBINE
Default	Default value for this parameter, for content type CT_LIST_SINGLE this may be one of the constants defined in Values	*
Values	For content type CT_LIST_SINGLE: list of possible values. Every value is described in a separate section of the inifile, holding its Tag and Description. For content type CT_EDIT: should be * to allow any value	

Sections for individual values

Parameter	Description	Example
Tag	Value constant used in report request text.	M
Description	Describes the value	Month

Translating Existing Reports

The language file TCRPT01.LNG holds the text constants used in the standard reports. You can create nationalized versions of the reports by translating this language file. TCRPT01.LNG resides in the Shared directory on the Report Agent workstation.

You can also translate the INI file of the report and the language file of the Report Request Wizard (TCPRepReq01.LNG on the client computers) in order to create a nationalized version of the Report Request panel in TC Management Console.

Recommendations for CSV Reports

Although the standard reports delivered with TC/Report support CSV output, the resulting CSV files will not match the needs of a connected billing application.

CSV files created by Crystal Reports contain much more information than needed by a billing system, e.g. the report header is repeated for almost every result line, groupings and group headers make the output more complicated etc.

We recommend to create an XLS report, use MS Excel to modify the resulting report according to the format needs of the billing application and save it in CSV format.

Chapter 9

Hints

This chapter describes hints for using Fetch Agent.

Do Not Change the SQL Query for the Reports

The Crystal Reports Designer creates an SQL query, based on used tables and on the record selection formula. Although the Report Designer allows you to view the SQL query and to edit it manually, please do not change it. Adapt the record selection formula instead.

How to Count the Number of SMS Sent

If you send 1 SMS which is split into two or more SMS, there is nevertheless only one TCOSS send order. If you then make a report, you see only this one send order and not the 2 or more.

To overcome this missing information, you can set within TC/Link-WM the configuration value *CostsPerSMS* = 1.

Then you see according to the cost value, how many individual SMS are sent.

Event Log Warning with SQL 2005

TC/Report makes heavy use of a Stored Procedure called SHRINK_LOG that is installed by KCS Setup. The purpose of this procedure is to eliminate the transaction log records, which otherwise can make the database too big after bulk changes (e.g. remove oldest records, adapt after new channel definition).

With SQL Server 2005, every call of SHRINK_LOG produces a warning in the application event log, stating that “BACKUP LOG WITH TRUNCATE_ONLY or WITH NO_LOG is deprecated. The simple recovery model should be used to automatically truncate the transaction log.”

To avoid this event log message, run TC/Report Fetch Agent Setup and in “Database settings for Fetch Agent” set option “Fetch Agent truncates transaction log” to “NO”.

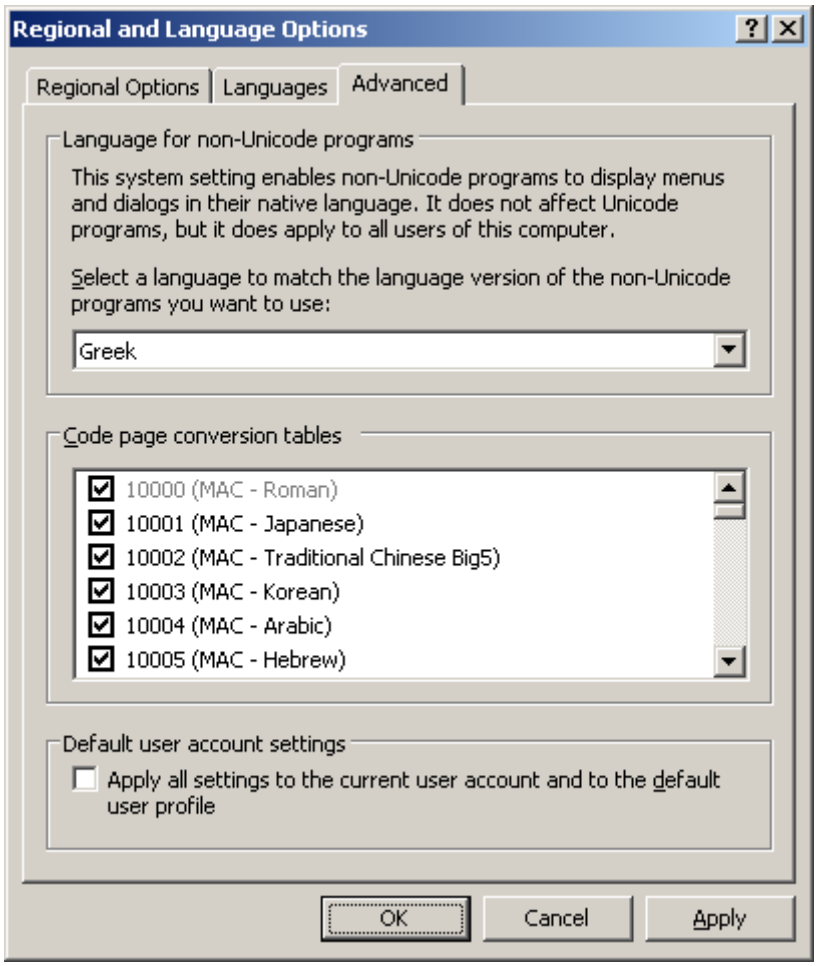
Let Setup update the database, it installs a dummy version of SHRINK_LOG that does not truncate the transaction log. For SQL Server version 2008 or higher, this is always the case.

See Fetch Agent Setup for details.

Using TC/Report with Non-Standard Languages and Character Sets

For installations with non-standard TCOSS character sets (Eastern-Europe, Japanese, Chinese, Greek), take care to configure the Regional and Language Settings on the TC/Report workstation accordingly.

Make sure that the correct language is set in the field "language version for the non-Unicode programs" in the Advanced panel.



Make sure that this configuration is done for the Windows account that TC/Report uses. If TC/Report uses the System Account, enable the checkbox “Apply all settings to the current user account and to the default user profile.”

TC/Report will choose a character set conversion that matches these local workstation settings. It is therefore not necessary to configure the TCOSS codepage.

Nevertheless, it is important that the database uses the correct character sets as well. The following section describes how to use an English SQL server for non-standard character sets.

Configuring the collation for the TC/Report database

With the English version of MS SQL Server 200X, it is possible to define the collation (character set and comparison style) per database.

TC/Report Setup (and also TC/Probe Setup) let you enter the collation that shall be used for the TC/Report database.

Please note that this only has effect when creating a new database. If the database already exists, the collation will not be changed.

For databases created with KCS Setup version 9.2 or later, string columns are defined as nvarchar. Therefore, the database collation affects only sort order and sensitivity.

The SQL Server Books Online holds more information about collations. This section only mentions settings that can be used with TC/Report.

Syntax for collation names:

CollationDesignator_CaseSensitivity_AccentSensitivity[_KanatypeSensitive [_WidthSensitive]] | _BIN

CollationDesignator: Defines the alphabet or language whose sorting rules are applied when dictionary sorting is specified, and the code page that is used for storing textual data.

CaseSensitivity: **CI** specifies case-insensitive, **CS** specifies case-sensitive.

AccentSensitivity: **AI** specifies accent-insensitive, **AS** specifies accent-sensitive.

KanatypeSensitive: **Omitted** specifies kanatype-insensitive, **KS** specifies kanatype-sensitive.

WidthSensitivity: **Omitted** specifies width-insensitive, **WS** specifies width-sensitive.

BIN: Specifies the binary sort order to be used.

To get a complete list of collations supported by SQL Server, run the following TSQL query:

SELECT Name, Description FROM fn_helpcollations()

Configurable String Separator for CSV Report Format

In CSV reports, strings are enclosed by double quotation marks, e.g. “sample value”.

This can be changed by creating a registry value HKLM\Software\Topcall\TCReport_Report\General \CsvStringSeparator of type REG_SZ.

If this registry value exists and is empty, no quotation marks will be used for string values in CSV reports. Alternatively, you can choose a different separator value, e.g. single quotation marks (CsvStringSeparator=’).

Failure Audit Log Entry During Report Creation

If auditing of login failures is enabled for the SQL server, a failure audit entry will be written to the application event log whenever a report is created.

Event Type: Failure Audit

Event Source: MSSQLSERVER

Event Category: (4)

Event ID: 18456

These log entries can be ignored. TC/Report intentionally attempts a wrong login to check the server connection.

Note You can disable login auditing in the SQL server configuration via the management application of the SQL server (MS SQL Server Management Studio. To disable login auditing, open the server properties, select the Security panel and set the audit level to “None”.

Calculated Recordsizes in Fetch Agent Trace

At a trace level of 255 or above, the Fetch Agent writes trace lines with information about the calculated average record size (in bytes) for the most important tables. This is only done if size limits are configured (either for the TCREPORT database itself or for the backup databases.

Example:

```
15/10:14:12.568 (790/1b9c) Action_Table rowsize: 811
15/10:14:12.568 (790/1b9c) Log_Table rowsize: 362
15/10:14:12.569 (790/1b9c) User_Table rowsize: 512
15/10:14:12.569 (790/1b9c) Probe_Table rowsize: 0
15/10:14:12.569 (790/1b9c) App_Downtime_Table rowsize: 113
15/10:14:12.615 (790/1b9c) Server_Table rowsize: 1005
15/10:14:12.618 (790/1b9c) ProbeDefinition_Table rowsize: 0
15/10:14:12.621 (790/1b9c) DueProbe_Table rowsize: 0
15/10:14:12.624 (790/1b9c) ProbeEndpoint_Table rowsize: 0
15/10:14:12.627 (790/1b9c) ProbeDefTarget_Table rowsize: 0
15/10:14:12.629 (790/1b9c) ProbeHistory_Table rowsize: 0
15/10:14:12.632 (790/1b9c) ID_Table rowsize: 0
15/10:14:12.648 (790/1b9c) Workstation_Table rowsize: 805
15/10:14:12.687 (790/1b9c) Application_Table rowsize: 91
15/10:14:12.715 (790/1b9c) AppGroup_Table rowsize: 143
15/10:14:12.755 (790/1b9c) App_AppGroup_Table rowsize: 21
15/10:14:12.790 (790/1b9c) Regkeys2TCApplication rowsize: 74
```

Chapter 10

Performance

Test environment:
TC/Report Workstation:
CPU: Pentium III, 500 MHz, bus frequency 100 MHz
Memory: 128 MB, 100 MHz
Hard disk capacity: 8 GB
Network: TCP/IP, 3Com 3C905B-TX, Ethernet, 10baseT Twisted pair, 100M/bit
Software: MS Windows NT 4.0 SP 6a, MS SQL Server 7.0 SP 2, TC/Report 1.03.00

TCOSS Server:
CPU: Pentium III, 500 MHz, bus frequency 100 MHz
Memory: 128 MB, 100 MHz
Hard disk capacity: 8 GB
Network: TCP/IP, 3Com 3C905B-TX, Ethernet, 10baseT Twisted pair, 100M/bit
Software: MS Windows NT 4.0 SP 6a, TCOSS 7.41.00

Fetch Agent performance:
New fetch, terminated send orders only (no send attempts, no log entries).
Number of messages per hour: 172100
Average CPU usage (new fetch): 63 %
Average CPU usage (idle mode): 0.8%
Memory usage: 9 – 10 MB

Report Agent performance:
The following report request (yielding a report with 274 data rows) was processed:

```
reportname = UNIV
DateFrom = 20010301:000000
DateTo = 20010331:235959
server_id = TOM
period = Month
wait = YES
grouporder = Server_ID,Time,Recipient,CostCenter
resultorder = SendOrders,Size,Cost
costcenter = FAX,VOICE,(*)
destination = FILE, D:\tcoss\Month%y%m.doc
resolution = m
internalrecipient = YES
outputformat = doc
```

```
faxprefix = 2
```

Number of reports per hour: 564
Average CPU usage (report generation): 54 %
Average CPU usage (idle mode): 8.1 %
Memory usage: minimum 9 MB, maximum undefined

Chapter 11

Results of Performance Tests on a TCOSS Server

Some customers may prefer to have all KCS applications on 1 server, thus running TC/Report on the TCOSS server itself. Although we do not recommend this type of installation, because of its implications on the TCOSS performance, we did some tests in such an environment. The following test results should help you in your decision.

Used Test Environment and Modus

This chapter describes the environment used for test and modus.

Environment

All test were done on an Intel Pentium II 233 MHz machine with a physical memory of 130488 K. On this machine, the following applications were installed: TCOSS 7.41.00, TC/POP3, and 2 instances of TC/LinkSC.

For the test, we installed the MSDE database system and updated it with SQP70sp2i.exe as described in the TC/Report manual. Next TC/Report was installed consisting of fetch agent and report agent. The fetch agent was configured to fetch data directly from the local TCOSS and put it into the locally installed database. Fetch delay was set to 20 ms and the poll cycle to 5s.

We set the report agent to a polling interval of 1 minute.

Modus

The tests focused on getting data on how much sending activity of TCOSS was disturbed when a locally installed fetch- and/or report agent are running. Supplementary we wanted to find out how much influence the possible different configurations of MSDE can have on the performance of the system.

To achieve this goal we simulated the sending of faxes on TCOSS by sending messages to NULL TUM's with a delay of 12 sec before sending the message, 3 sec before sending a page and 50 msec before sending a line. The exact testing procedure was the same for each single test:

1000 send orders to a NULL TUM - as described above – were created on TCOSS with a specific time for sending and a time for latest delivery exactly 10 minutes later, for a message containing of 2 pages of TCI code and a standard coversheet containing all possible coversheet variables. No other messages were sent within that 10 minute time period. After that 10 minutes had elapsed, we counted how many of that 1000 send orders were successfully sent.

While TCOSS was sending the agents of TC/Report were running in several modi:

No TC/Report Activity at All

Some test were done with report and fetch agent disabled. This test gave us the number of messages sent within 10 minutes without any interference by TC/Report.

Fetch Agent Fetching

In this test modus report and fetch agent were both running. We sent no requests for reports so report agent simply polled for reports once per minute but did not have any reports to create. Fetch agent had already fetched all messages from TCOSS and was only updating the database with new messages being currently sent.

Fetch Agent Refetching

In this test modus report and fetch agent were both running. We sent no requests for reports so report agent simply polled for reports once per minute but did not have any reports to create. In the TC Management Console we clicked on refetch, thus clearing the database and refetching all entries from TCOSS into the database.

Report Agent Continuously Reporting

In this test modus report and fetch agent were both running. We sent 30 report requests with a starting time for sending 2 minutes before the 1000 messages were sent to the NULL Tums. The latest delivery time was set to 2 minutes after the latest delivery time of the test messages.

Database Updating

Directly after we clicked on refresh the fetch agent showed in the KCS Monitor the status database updating. We also did a few tests when fetch agent was in this status. During this time report agent was running but we requested no reports.

Test Results

This chapter describes the test results.

Configuration and Results

The test showed that the standard settings of MSDE, as it is configured after installation, have the least influence on TCOSS performance. It may be only necessary to enlarge the '*max server memory*' setting if your database is very large.

The following table shows the configuration of our MSDE after installation with the only exception that '*max server memory*' was increased from 80 to 160 MB and '*max worker threads*' was set to 1024.

name	minimum	maximum	config_value	run_value
affinity mask	0	2147483647	0	0

allow updates	0	1	0	0
cost threshold for parallelism	0	32767	5	5
cursor threshold	-1	2147483647	-1	-1
default language	0	9999	0	0
default sortorder id	0	255	52	52
extended memory size (MB)	0	2147483647	0	0
fill factor (%)	0	100	0	0
index create memory (KB)	704	1600000	0	0
language in cache	3	100	3	3
language neutral full-text	0	1	0	0
lightweight pooling	0	1	0	0
locks	5000	2147483647	0	0
max async IO	1	255	32	32
max degree of parallelism	0	32	0	0
max server memory (MB)	4	2147483647	160	160
max text repl size (B)	0	2147483647	65536	65536
max worker threads	10	1024	1024	1024
media retention	0	365	0	0
min memory per query (KB)	512	2147483647	1024	1024
min server memory (MB)	0	2147483647	0	0
nested triggers	0	1	1	1
network packet size (B)	512	65535	4096	4096
open objects	0	2147483647	0	0
priority boost	0	1	0	0
query governor cost limit	0	2147483647	0	0
query wait (s)	-1	2147483647	-1	-1
recovery interval (min)	0	32767	0	0
remote access	0	1	1	1
remote login timeout (s)	0	2147483647	5	5
remote proc trans	0	1	0	0
remote query timeout (s)	0	2147483647	0	0
resource timeout (s)	5	2147483647	10	10
scan for startup procs	0	1	0	0
set working set size	0	1	0	0
show advanced options	0	1	1	1
spin counter	1	2147483647	10000	0
time slice (ms)	50	1000	100	100
two digit year cutoff	1753	9999	2049	2049
Unicode comparison style	0	2147483647	196609	196609
Unicode locale id	0	2147483647	1033	1033
user connections	0	32767	0	0
user options	0	4095	0	0

The *minimum* and the *maximum* column show the minimum and the maximum value a single option can have. The *config_value* column shows the value that is currently configured. But the currently configured value need not be the value actually used by the database because sometimes it is necessary to reboot the MSDE before a new setting is used by it. The column *run_value* shows the settings currently used by the database.

You get this config table of MSDE by connecting to the database with *osql.exe*. Then you have to execute the statements:

sp_configure
go

To change the configuration of an option you have to execute:

sp_configure ‘option_name’, [new value]
go
reconfigure

go

Next you should call *sp_configure* again and check if the new value is only displayed in the *config_value* column or also in the *run_value* column. If it is only in the *config_value* column you have to reboot MSDE otherwise the new value for that option is not used by the database.

Results Sorted by Database Options

As explained above we changed some database options during testing so that we could compare their influence on TCOSS performance. This section presents this influence for each of that options by listing in each table how many messages were sent from TCOSS within 10 minutes.

Abbreviations used in the tables:

w. Threads	-‘maximum number of working’ threads option of MSDE
sent. M.	- messages sent within the interval of 10 minutes
Chann.	- number of Null TUM channels configured on TCOSS
min Mem	- ‘minimum Memory per query (MB)’ option of MSDE
MaxDBSize	‘maximum server memory (MB)’ option of MSDE
c-report	- Report agent continuously reporting
fetch	- Fetch agent fetching
refetch	- Fetch agent refetching
updating	- Database updating
nofetch	- No TC/Report activity at all

Max Working Threads

This option can be changed without rebooting MSDE.

w. Threads	Phase	sent M.	Chann.	min. Mem	Fill Factor	MaxDBSize	priority boost
1024	c-report	463	50	1024	0	160	FALSE
10	c-report	450	50	1024	0	160	FALSE
10	c-report	566	80	1024	0	160	FALSE
1024	c-report	574	80	1024	0	160	FALSE
1024	fetch	502	50	1024	0	160	FALSE
10	fetch	545	50	1024	0	160	FALSE
10	fetch	550	50	1024	0	160	FALSE
1024	fetch	790	80	1024	0	160	FALSE
1024	fetch	822	80	1024	0	160	FALSE
10	fetch	740	80	1024	0	160	FALSE
1024	nofetch	550	50	1024	0	160	FALSE
1024	nofetch	550	50	1024	0	160	FALSE
1024	nofetch	550	50	1024	0	160	FALSE
1024	nofetch	840	80	1024	0	160	FALSE
10	nofetch	828	80	1024	0	160	FALSE
10	refetch	550	50	1024	0	160	FALSE
1024	refetch	550	50	1024	0	160	FALSE
1024	refetch	540	50	1024	0	160	FALSE
1024	refetch	511	50	1024	0	160	FALSE
1024	refetch	452	50	1024	0	160	FALSE
10	refetch	751	80	1024	0	160	FALSE
1024	refetch	738	80	1024	0	160	FALSE
1024	refetch	736	80	1024	0	160	FALSE
10	updating	332	50	1024	0	160	FALSE
1024	updating	501	50	1024	0	160	FALSE

Max worker threads set to 1024 gives a slightly better performance, the difference in performance is high in updating mode.

Max Server Memory (MB)

This option can be changed without rebooting MSDE.

MaxDBSize	Phase	sent M.	Chann.	w. Threads	min Mem.	Fill Factor	priority boost
4	c-report	564	80	1024	1024	0	FALSE
160	c-report	574	80	1024	1024	0	FALSE
4	fetch	643	80	1024	1024	0	FALSE
160	fetch	790	80	1024	1024	0	FALSE
160	fetch	822	80	1024	1024	0	FALSE
4	nofetch	821	80	1024	1024	0	FALSE
160	nofetch	840	80	1024	1024	0	FALSE

A max DBSize of 4 leads to only 80% performance of a max DBSize of 160 MB. Setting the upper limit for the database size to low leads to performance loss.

Priority Boost

After changing this option you have to reboot MSDE so that it gets in effect.

priority boost	Phase	sent M.	Channels	w. Threads	min. Mem	Fill Factor	MaxDBSize
TRUE	c-report	568	80	1024	1024	0	160
FALSE	c-report	574	80	1024	1024	0	160
TRUE	fetch	644	80	1024	1024	0	160
FALSE	fetch	790	80	1024	1024	0	160
FALSE	fetch	822	80	1024	1024	0	160
FALSE	nofetch	840	80	1024	1024	0	160

Setting the priority boost option drops the number of sent messages within 10 minutes by 19%.

Fill Factor

Changing this option requires MSDE to be rebooted so that it gets in effect.

Fill Factor	Phase	sent M.	Chann.	min Mem.	w. Threads	MaxDBSize	priority boost
0	c-report	581	80	512	1024	160	FALSE
10	c-report	513	80	512	1024	160	FALSE
0	c-report	574	80	1024	1024	160	FALSE
10	c-report	561	80	1024	1024	160	FALSE
10	c-report	640	80	1024	1024	160	FALSE
0	c-report	560	80	4096	1024	160	FALSE
10	c-report	541	80	4096	1024	160	FALSE
0	fetch	802	80	512	1024	160	FALSE
10	fetch	644	80	512	1024	160	FALSE
0	fetch	790	80	1024	1024	160	FALSE
0	fetch	822	80	1024	1024	160	FALSE
10	fetch	708	80	1024	1024	160	FALSE
10	fetch	767	80	1024	1024	160	FALSE
10	fetch	647	80	1024	1024	160	FALSE
10	fetch	681	80	1024	1024	160	FALSE
10	fetch	633	80	1024	1024	160	FALSE
0	fetch	743	80	4096	1024	160	FALSE
10	fetch	640	80	4096	1024	160	FALSE
0	nofetch	840	80	1024	1024	160	FALSE
10	nofetch	844	80	1024	1024	160	FALSE

In every phase, taken with any combination of minimum Memory per Query a Fill Factor of 0 has a better performance than a Fill Factor of 10. Therefore this value should be prefered.

Minimum Memory per Query (in MB)

This option can be changed without rebooting MSDE.

min. Mem	Phase	sent M.	Chann.	Fill Factor	w. Threads	MaxDBSize	priority boost
512	c-report	581	80	0	1024	160	FALSE
512	c-report	513	80	10	1024	160	FALSE
1024	c-report	574	80	0	1024	160	FALSE
1024	c-report	561	80	10	1024	160	FALSE
1024	c-report	640	80	10	1024	160	FALSE
4096	c-report	560	80	0	1024	160	FALSE
4096	c-report	541	80	10	1024	160	FALSE
512	fetch	644	80	10	1024	160	FALSE
512	fetch	802	80	0	1024	160	FALSE
1024	fetch	708	80	10	1024	160	FALSE
1024	fetch	790	80	0	1024	160	FALSE
1024	fetch	822	80	0	1024	160	FALSE
1024	fetch	767	80	10	1024	160	FALSE
1024	fetch	647	80	10	1024	160	FALSE
1024	fetch	681	80	10	1024	160	FALSE
1024	fetch	633	80	10	1024	160	FALSE
4096	fetch	743	80	0	1024	160	FALSE
4096	fetch	640	80	10	1024	160	FALSE
1024	nofetch	844	80	10	1024	160	FALSE
1024	nofetch	840	80	0	1024	160	FALSE
1024	refetch	736	80	0	1024	160	FALSE
1024	refetch	738	80	0	1024	160	FALSE

The table shows that a smaller value of minimum Memory per Query should be preferred to a greater one when Fill Factor is set to 0. If Fill Factor is set to 10 min. Mem of 4096 is very similar to 512 in performance. We recommend a combination of Fill Factor 0 and a low minimum Memory.

Recommended Values for the Tested Database Settings

As already mentioned it seems to be best to leave the database settings as they are after installation of MSDE. Therefore we recommend the following values for the 5 tested database settings:

‘Max worker threads’	1024 or another high value
‘Fill factor’	0
‘Max server memory (MB)’	160, use a value which is surely higher than your databases memory need.
‘Priority boost’	0
‘Minimum memory per query’	1024, use a value near the minimum

Performance Loss Under Recommended Database Configuration

This section describes the performance losses for recommended database configuration.

On TCOSS with 80 Channels

Phase	sent M.	Chann.	w. Threads	min. Mem	Fill Factor	MaxDBSize	priority boost
c-report	574	80	1024	1024	0	160	FALSE
fetch	790	80	1024	1024	0	160	FALSE
fetch	822	80	1024	1024	0	160	FALSE
refetch	736	80	1024	1024	0	160	FALSE
refetch	738	80	1024	1024	0	160	FALSE
nofetch	840	80	1024	1024	0	160	FALSE

Performance in Percentage:

Nofetch: 100%

Fetch: 95%

Refetch: 88%

c-Report: 68%

On TCOSS with 50 Channels

Phase	sent M.	Chann.	w. Threads	min. Mem	Fill Factor	MaxDBSize	priority boost
c-report	463	50	1024	1024	0	160	FALSE
fetch	502	50	1024	1024	0	160	FALSE
refetch	511	50	1024	1024	0	160	FALSE
refetch	452	50	1024	1024	0	160	FALSE
nofetch	550	50	1024	1024	0	160	FALSE
nofetch	550	50	1024	1024	0	160	FALSE
nofetch	550	50	1024	1024	0	160	FALSE

Performance in Percentage:

Nofetch: 100%

Fetch: 91 %

Refetch 88 %

c-Report: 68%

Raw Test Results

Test ID	Phase	sent M.	Chann.	MaxDBSize	w.Threads	priority boost	Fill Factor	min. Mem
1	fetch	502	50	160	1024	FALSE	0	1024
2	refetch	452	50	160	1024	FALSE	0	1024
3	nofetch	550	50	160	1024	FALSE	0	1024
4	fetch	822	80	160	1024	FALSE	0	1024
5	nofetch	840	80	160	1024	FALSE	0	1024
6	refetch	738	80	160	1024	FALSE	0	1024
7	report	699	80	160	1024	FALSE	0	1024
8	nofetch	550	50	160	1024	FALSE	0	1024
9	fetch	545	50	160	10	FALSE	0	1024
10	fetch	550	50	160	10	FALSE	0	1024
11	updating	332	50	160	10	FALSE	0	1024
12	updating+fetch	350	50	160	1024	FALSE	0	1024
13	refetch	550	50	160	10	FALSE	0	1024
14	c-report	450	50	160	10	FALSE	0	1024
15	c-report	463	50	160	1024	FALSE	0	1024
16	refetch	550	50	160	1024	FALSE	0	1024
17	nofetch	550	50	160	1024	FALSE	0	1024
18	updating	501	50	160	1024	FALSE	0	1024
19	refetch	540	50	160	1024	FALSE	0	1024
20	refetch	736	80	160	1024	FALSE	0	1024
21	refetch	511	50	160	1024	FALSE	0	1024
22	fetch	790	80	160	1024	FALSE	0	1024
23	c-report-failed	729	80	160	1024	FALSE	0	1024
24	c-report	574	80	160	1024	FALSE	0	1024
25	c-report	566	80	160	10	FALSE	0	1024
26	fetch	740	80	160	10	FALSE	0	1024
27	nofetch	828	80	160	10	FALSE	0	1024
28	refetch	751	80	160	10	FALSE	0	1024
29	fetch	644	80	160	1024	TRUE	0	1024
30	c-report	568	80	160	1024	TRUE	0	1024
31	c-report	564	80	4	1024	FALSE	0	1024
32	fetch	643	80	4	1024	FALSE	0	1024
33	nofetch	821	80	4	1024	FALSE	0	1024
34	c-report	561	80	160	1024	FALSE	10	1024
35	c-report	640	80	160	1024	FALSE	10	1024
36	fetch	767	80	160	1024	FALSE	10	1024
37	fetch	708	80	160	1024	FALSE	10	1024
38	fetch	640	80	160	1024	FALSE	10	4096
39	c-report	541	80	160	1024	FALSE	10	4096
40	fetch	644	80	160	1024	FALSE	10	512
41	c-report	513	80	160	1024	FALSE	10	512
42	fetch	647	80	160	1024	FALSE	10	1024
43	nofetch	844	80	160	1024	FALSE	10	1024
44	fetch	681	80	160	1024	FALSE	10	1024
45	fetch	633	80	160	1024	FALSE	10	1024
46	fetch	743	80	160	1024	FALSE	0	4096
47	c-report	560	80	160	1024	FALSE	0	4096
48	c-report	581	80	160	1024	FALSE	0	512
49	fetch	802	80	160	1024	FALSE	0	512

Chapter 12

Restrictions and Significant Limitations

General Restrictions of the Software

- **No parallel operation**:

  – only 1 Fetch Agent per database is supported

  – Fetch Agent and Report Agent use fixed registry paths for their configuration. Therefore parallel instances on the same computer are not supported.

Database

- The database structure is documented; **direct read-only access to the database** by customer applications is allowed.
- The **maximum size of backup databases** is considered only as a hint. The real size can be slightly higher or lower than the configured maximum. Additionally, the size of the transaction log file can increase dramatically during certain operations, e.g. selecting “Refetch” or “Delete” in TC Management Console, creating a backup database, channel update. The Fetch Agent shrinks the log file afterwards. Nevertheless, these operations need hard disk space.
- It is not allowed to change the structure of tables, indexes, views and stored procedures that were created by TC/Report setup or other components of TC/Report. TC/Report relies on a defined table structure and is not guaranteed to work if you change this structure (e.g. by adding columns, configuring partitions, configuring compression etc.)

User Interface

- Just **English language files** are delivered for TC Management Console panels of TC/Report.
- To enable correct display of all characters of the UTF-16 range seen in the TCMC, you must configure a font that supports Unicode, for instance “Arial Unicode MS”. This is done in TCMC - Common Settings – Workstation.

Fetch Agent

- Only 1 Fetch Agent instance (including FetchWorkStations.exe) per database is allowed.
- Only 1 Fetch Agent instance (including FetchWorkStations.exe) can run on a single computer.
- The stored **message size** can slightly differ from the true value (e.g. Cover sheets are not counted as part of the message size)
- Each **channel** can have only one **message class**. If you change the Channel – Message Class assignment, it will be done for each entry in the database. E.g. if a channel is assigned to Fax and it won’t be needed anymore, this channel should not be used for other services.
- The LINES report and certain features of the UNIV report (group by country code or channel group) only work if **physical channels** and channel groups are configured in a special way (___ syntax in

TC Management Console). Only entries fetched after configuring the channel or channel group are evaluated.

- Calculation only based on existing data where **cost accounting** was activated, not for data of the past (e.g. if CostPerRecipient for TC/LINK's is activated after installing TC/REPORT, the reported costs for the TC/LINK's are only valid from this moment on).
- In the case of a (very unlikely) **desync condition of a KCS Tandem server** logging entries can get lost.
- **Backup: "send a message to user"** sends to first enabled TCOSS server
- The current version of TCOSS does not yet log the usage of the **TC/VoiceAccess**. Therefore, these activities cannot be part of a report.
- When triggering an event or sending a notification, TCOSS copies custom field values from the original send order into the new send order. If the event or notification is archived, be careful to interpret the information properly (use the parameters WithEvents and WithNotifs in UNIV report to filter for original messages).
- Custom field values may get truncated by TCOSS. Each field can have 128 characters, but the total length of all four custom fields is limited to 320 characters.
- If TCSNMP is installed on the Fetch Agent computer, it is recommended to use ASCII strings for the logical TCOSS server names configured in the TCMC Servers panel. TCSNMP encodes characters that are not part of the local Windows code page as hex digits, thus making the names unreadable. This encoding makes the string longer. For compliance with the SNMP protocol, TCSNMP truncates encoded strings that are longer than 100 bytes. This means that pure Unicode names can be truncated to 12 characters.
  More information about encoding can be found in the TCSNMP manual, section "TC/SNMP Advanced Settings".

Report Agent

- Only one instance of the Report Agent can run on a single computer.
- The **error code QR** used by TC/Report is reported as "undefined error code" in KCS link versions below 2.03.02 and KCS Client Applications versions below 5.00.07. Solution: Use the newest TC0x.ERR files.
- **Crystal reports can access only one database** – when the data is split in more than one (backup) databases you have to split the request.
- If you want to fax a report, use the syntax **„Destination=FAX,number"**. Although the option „Destination=TOPCALL,FAX,number" is syntactically correct, it cannot be used because it would send a binary file to the fax channel.
- When a report runs into an error, the report agent sends an error message. Depending on the severity of the error, the report is terminated or retried. Periodic reports are only stopped if a severe error occurs (e.g. not existing report name specified).
  The text of the error message does not state whether a periodic report has been stopped or not.
  The TC/Report User Manual, section Errors / Troubleshooting, contains a detailed list about possible errors and their handling for periodic and non-periodic reports.
- With previous versions of Crystal Reports, there could be memory leaks when creating reports. Therefore, a mechanism to restart the Report Agent periodically was implemented. This mechanism is still part of the product: the Report Agent is restarted after 1000 report requests (configurable).
- **A printer must be installed on the Report Agent computer**. Otherwise, the generated report files are in portrait layout instead of landscape.

Reports

- **No borders in DOC reports**. (general Crystal report errors and restrictions apply).
- Currently creation of **new reports** or modifications of reports is only supported via **Professional Services**.
- Only reports that are part of the current KCS are actively supported (e.g. tested with a new release).
- The **scale of graphs** in graphic reports is determined automatically be Crystal Reports. For some customer requirements, it might be necessary to use a fixed scale instead. This must be changed manually via the Crystal Reports Designer.
- Grouping category values longer than 27 characters can become invisible when exporting to RTF format.
- In DOC and RTF reports, server names with non-ASCII characters can be truncated at the bottom. The report fields are too small to hold e.g. Chinese or Arabic characters. A workaround would be to manually increase the field size.
- In reports sent to destination “PRINT” or “FAX”, text fields including simultaneously “left to right” and “right to left” characters are corrupted. E.g., if the text field consists of English and Arabic characters, the Arabic characters overwrite the English characters.

Chapter 13

Usage Example: An ASP Billing Interface

This ASP billing interface provides billing information of the KCS system in a defined XML-format. This interface is a customized solution that is not part of the standard TC/Report version.

Two different types of files are generated periodically, one daily and one monthly file. The location of the files can be configured in the registry.

Daily-generated files consist of information regarding number of in- and outbound SMS and FAX messages, number of inbound VOICE messages and duration of outbound FAX (split to different areas) and satellite FAX messages.

Monthly-generated files store information about the amount of used archive space per customer.

The user also has the possibility to generate files manually.

This example and other customer specific changes are available via Professional Services.

Installation

Some manual changes are necessary to activate the TCReportXML application.

Install Additional Files

Copy the flowing files into the directory C:\TOPCALL:

TCReportXML_GenerateArchiveTables.sql
TCReportXML_ODBC_Registry.reg
TCReportXML_Registry.reg
TCReportXML.exe

Create Output Directories

Create the file destination folders (can be configured in the registry):

C:\TOPCALL\Billing

C:\TOPCALL\Billing\Automatic

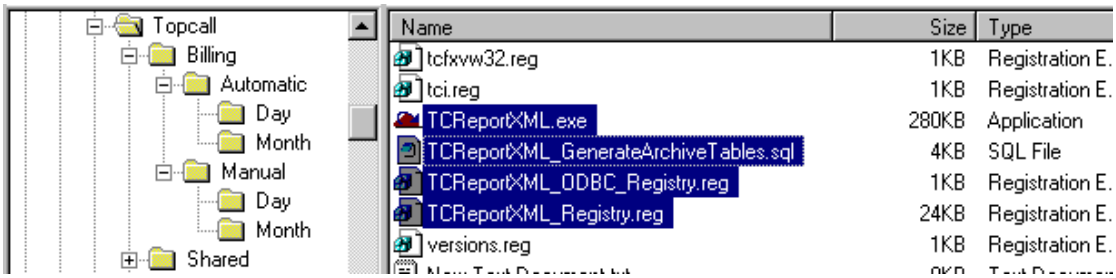
C:\TOPCALL\Billing\Automatic\Day

C:\TOPCALL\Billing\Automatic\Month

C:\TOPCALL\Billing\Manual

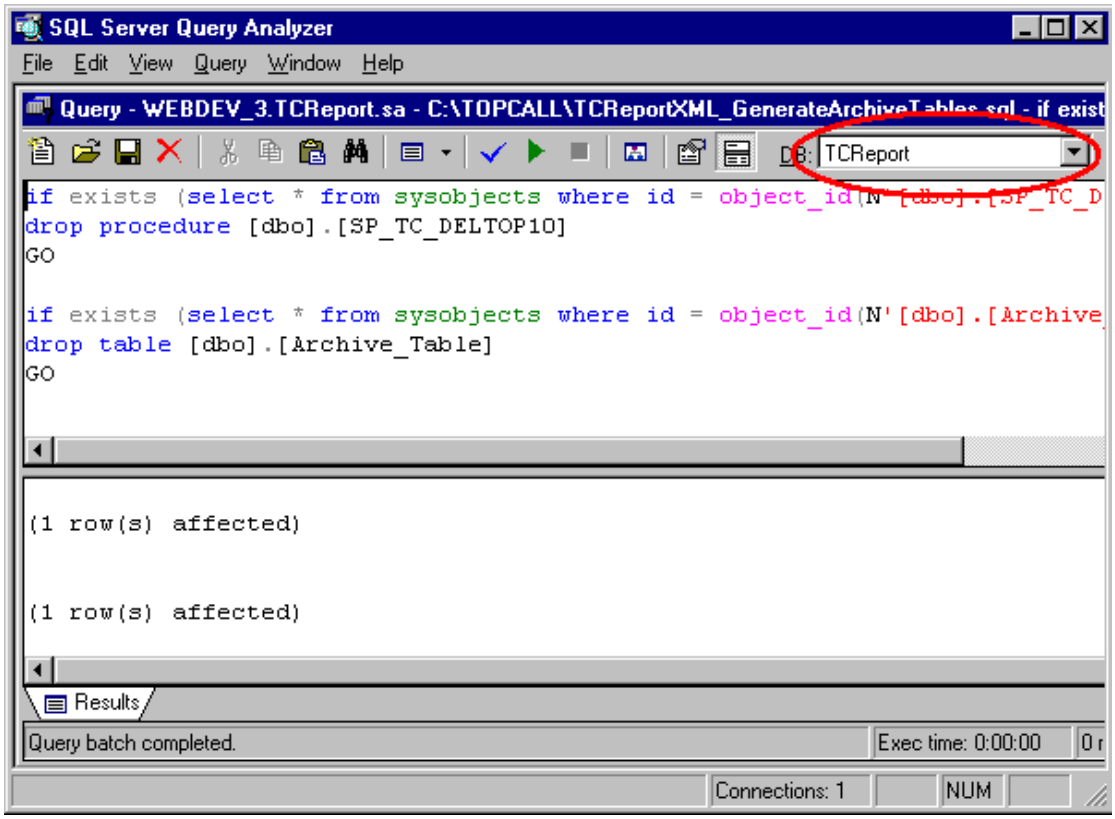
C:\TOPCALL\Billing\Manual\Day

C:\TOPCALL\Billing\Manual\Month

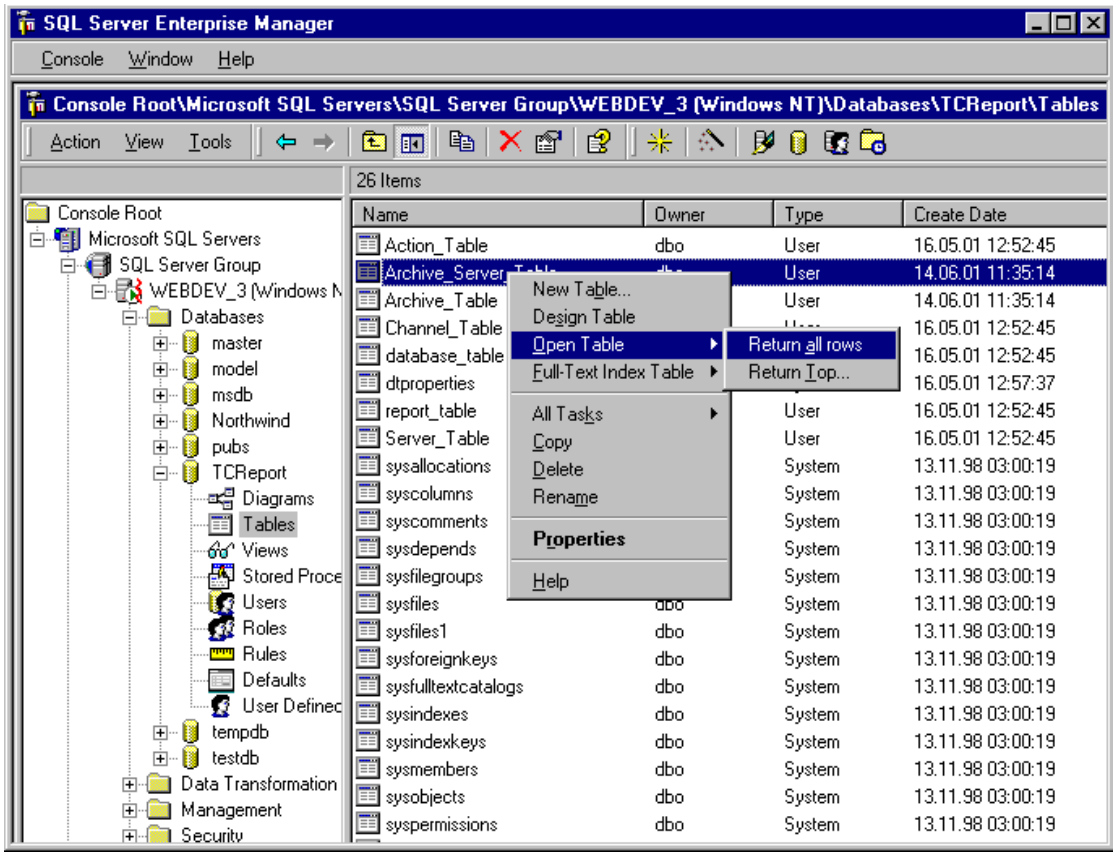


Generate and Configure TCReportXML Table

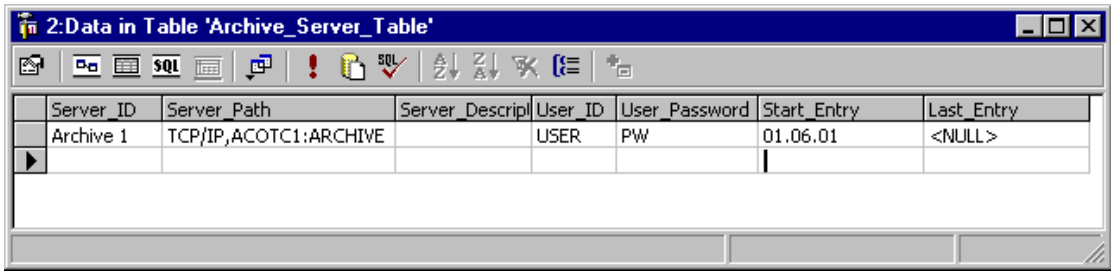
To generate the additional tables in the database, start the ‘SQL-Query Analyzer’ and connect to your SQL-Server. Then open the file ‘C:\TOPCALL\ TCReportXML_GenerateArchiveTables.sql’, select the database ‘TCReport’ and execute the script.



After creating the new tables, you have to configure the connection to the KCS Archive Server. Therefore open the ‘Archive_Server_Table’ in the ‘TCReport ’ database:



Here you have to configure the KCS Archive Server connection:



Server_Id: used in KCS Monitor to identify the Server

Server_Path, User_ID, User_Password: Connection to Server

Server_Description: not used

Start_Entry: date to start archive logging (if it is very long ago, logging will take very long until it is up to date!

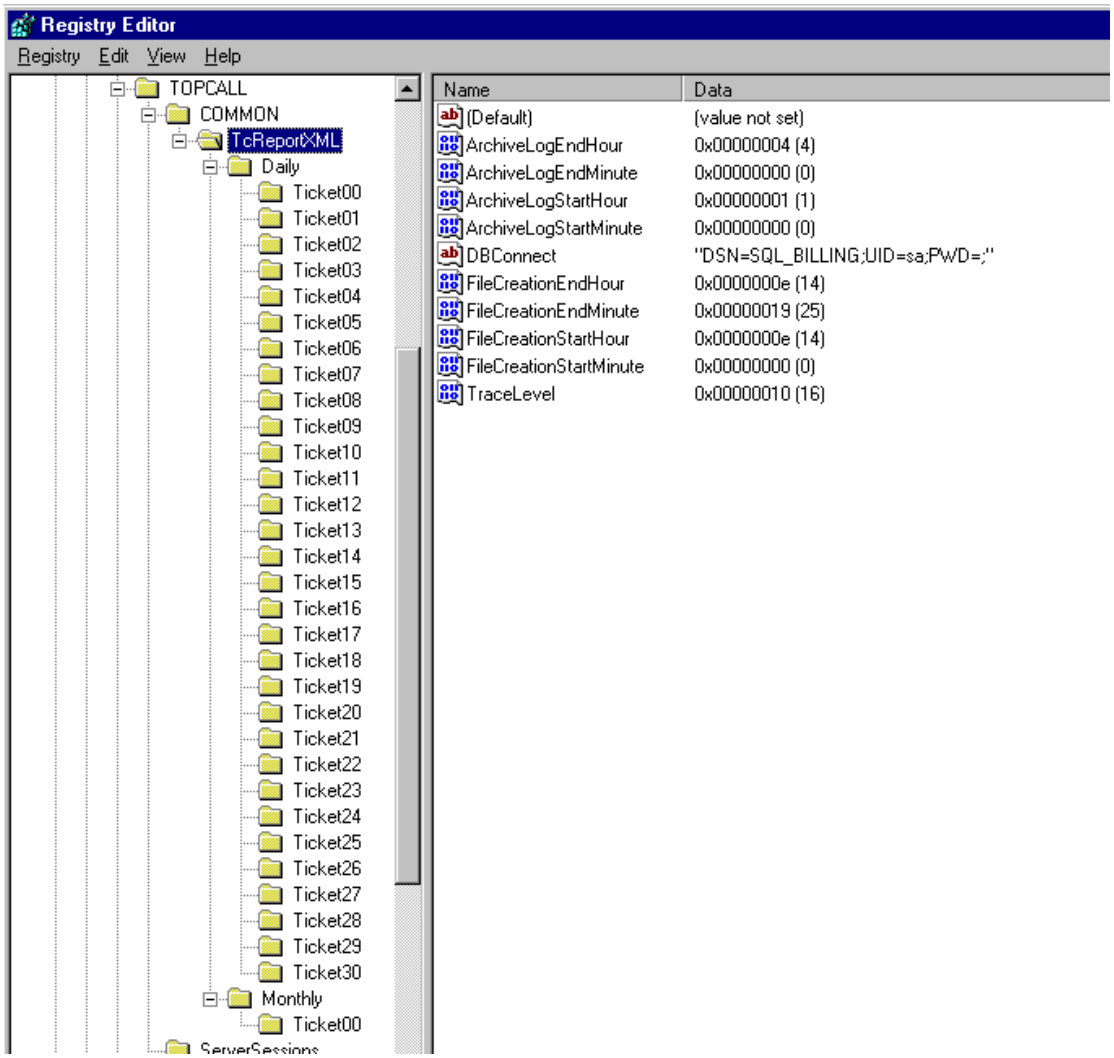
Last_Entry: leave '<NULL>'

Create TCReportXML ODBC connection

To create the ODBC connection to the ‘TCReport’ database, double-click the file ‘C:\TOPCALL\TCReportXML_ODBC_Registry.reg’ in the explorer window.

Modify the Registry for TCReportXML

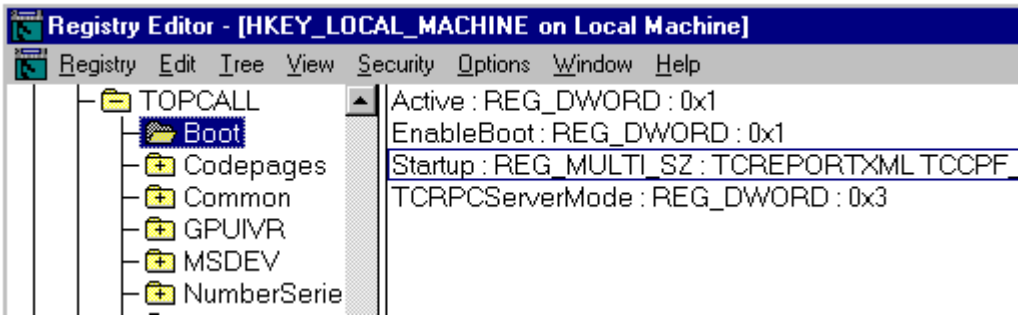
To enter the default values into the registry double-click the file ‘C:\TOPCALL\TCReportXML_Registry.reg’ in the explorer window.



For the meaning of the values, see the chapter Registry.

Modify the Registry for TCSRV Support

In the TOPCALL\Boot key within the registry, add the string ‘TCReportXML’ to the Startup value. This enables TCSRV and KCS Monitor to control the TCReportXML application.



Registry

There are two directories where the Registry values are stored:

For manual file creation (not started from TCSRV):

```
'HKEY_CURRENT_USER\Software\TOPCALL\COMMON\TCReportXML'
```

For automatic file creation:

```
'HKEY_USERS\.DEFAULT\Software\TOPCALL\COMMON\TCReportXML'
```

TraceLevel (type: REG_DWORD, default: 0x10):

```
>= 0x80 all Traces
>= 0x20 operational outputs
>= 0x10 several warnings
```

DBConnect (type: REG_SZ, default: 'DSN=SQL_BILLING;UID=sa;PWD=;'):

```
ODBC connection string
```

ArchiveLogStartHour (type: REG_DWORD, default: 1, Range: 0-23):

```
Timeframe for Archive scanning (start hour)
```

ArchiveLogStartMinute (type: REG_DWORD, default: 0, Range: 0-59):

```
Timeframe for Archive scanning (start minute)
```

ArchiveLogEndHour (type: REG_DWORD, default: 3, Range: 0-23):

```
Timeframe for Archive scanning (end hour)
```

ArchiveLogEndMinute (type: REG_DWORD, default: 0, Range: 0-59):

```
Timeframe for Archive scanning (end minute)
```

FileCreationStartHour (type: REG_DWORD, default: 14, Range: 0-23):

```
Timeframe for file creation (start hour)
```

FileCreationStartMinute (type: REG_DWORD, default: 0, Range: 0-59):

```
Timeframe for file creation (start minute)
```

FileCreationEndHour (type: REG_DWORD, default: 14, Range: 0-23):

```
Timeframe for file creation (end hour)
```

FileCreationEndMinute (type: REG_DWORD, default: 25, Range: 0-59):

```
Timeframe for file creation (end minute)
```

For daily and monthly files exist a sub key with the following settings:

FileIndex (type: REG_BINARY, default: 0)

```
Running number in Filename
```

FilePath (type: REG_SZ, default: ‘C:\TOPCALL\Billing\’)

```
Path of the Filename
```

FileNamePrefix (type: REG_SZ, default: ‘Cstream_daily_usage_’)

```
Prefix of the Filename
```

Filename = FileNamePrefix + running number + FileNamePostfix

FileNamePostfix (type: REG_SZ, default: ‘.infocdr’)

```
Postfix of the Filename
```

RateCategory (type: REG_SZ, default: ‘3’)

```
Ratecategory for that file type
```

ServiceAddress (type: REG_SZ, default: ‘1.1.1.1’)

```
Serviceaddress for that file type
```

ServiceName (type: REG_SZ, default: ‘10800’/’10801’)

```
Servicename for that file type
```

ServiceStatus (type: REG_SZ, default: ‘1’)

```
Servicestatus for that file type
```

TicketInfoPrefix (type: REG_SZ, default ‘/juxto/cstream/’)

```
Prefix for the ticket info
```

TicketNumber (type REG_DWORD, default: 1 / 31)

```
Number of tickets for that file type
```

For each ticket (corresponding to the Ticketnumber) a sub key exists:

info (type: REG_SZ)

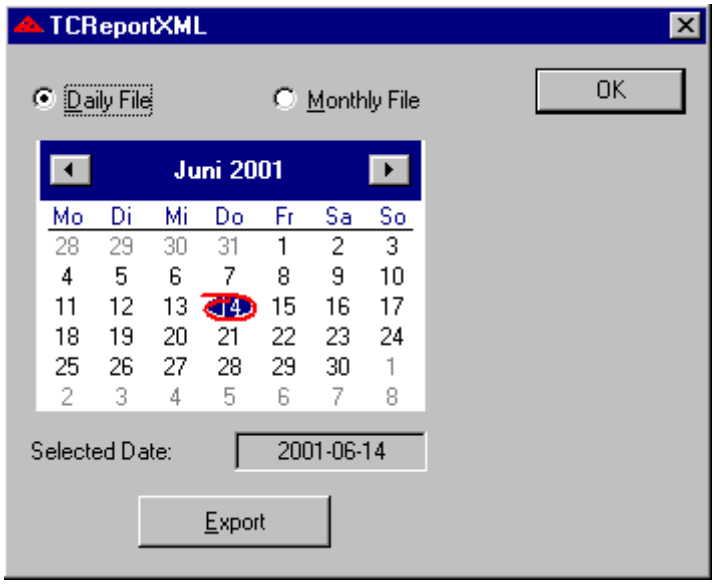
```
Ticketinfo for that ticket type
```

select (type REG_SZ)

```
SQL-Select statement to calculate the quantity for that ticket.
```

(‘WHERE’ clause for the time period and for the CostCenter will be added to this string by the TCReportXML application)

Manual File Creation



In this dialog, the user can select either a ‘Daily File’ or a ‘Monthly File’.

With the ‘Date-Control’, the wanted date can be selected (is indicated in the ‘Selected Date’ field).

Pressing the ‘Export’ button creates the file in the configured location.
The ‘OK’ button closes the TCReportXML application.

If an error occurs, by exporting a file, an error message box appears: “Creating file failed! Please check event log!”