Connecting Software
Connect Bridge – User Manual

Summary

This guide is designed to enable the reader to effectively work with the Connecting Software Connect Bridge (CB). It will introduce the various tools like the Connect Bridge Query Analyzer, the CB Administration Tool, the CB SQL Language Reference and many others.

Document History

Version	Date	Author	Changes
1.0	2013-04-12	PAN	Document creation
1.1	2015-06-18	MBE	Design adaption

Table of Contents

1 Overview 4
1.1 Document Structure 4
1.2 Operating Principles 5
2 CB Administration Tool 6
2.1 "Connect To" CB-Server 7
2.2 Description of the User Interface Components 9
2.3 Managing Logs 10
2.4 Managing Connectors 11
2.5 Managing Accounts 15
2.6 Managing Groups 18
2.7 Managing Users 21
3 The CB Service Controller 24
4 The CB Query Analyzer 26
4.1 Layout of the CB Query Analyzer 26
4.2 Working with the CB Query Analyzer 32
4.2.1 Connect 34
4.2.2 Execute a CB SQL statement or Stored Procedure 35
4.2.3 View Results/Errors 38
5 Workflow Development Tutorial 39
5.1 Scenario 39
5.1.1 C# sample 39
5.1.2 Java sample: 42
6 CB SQL Reference 45
6.1 General SQL Syntax Schema 45
6.2 Supported Data Manipulation Statements: 45
6.2.1 SELECT 45
6.2.2 INSERT 47
6.2.3 UPDATE 47
6.2.4 DELETE 48
6.2.5 STORED PROCEDURES 48
6.2.6 JOINS 49
6.2.7 CONDITIONS 50
6.2.8 FUNCTIONS 51
6.2.9 DATA TYPES 53

7 Glossary and Abbreviations 54
7.1 Glossary 54
7.2 Abbreviations 56

1 Overview

This guide is designed to give a more in depth view of the Connect Bridge components (and how to use them) then the Installation & Configuration Guide. The aspects below will be covered by this manual.

WARNING! Former name of the company and product has been modified from 'CNS Connect' "Media Gateway" to 'Connecting Software' & "Connect Bridge" respectively. This is why several file-names include the abbreviation "MG" or "MGW". If you should meet the term "Media Gateway" while using the Connect Bridge, please kindly email us a screenshot.

Thanks for understanding. Connecting Software documentation team.

1.1 Document Structure

The CB Administration Tool

Chapter 2 provides an in depth look into the functionality of the administration tool provided by Connecting Software. It covers managing of Accounts, Groups, Users, Licenses, Logs, etc.

The CB Service Controller

Chapter 3 contains a brief introduction to the service controller. A practical little tool designed to handle common administrative issues without having to resort to the Admin Tool or Windows Administration.

The CB Query Analyzer

Chapter 4 covers the Query Analyzer. Connecting Software's main tool to query & test the functionality of the CB Server by means of the CB SQL Language.

A Brief Workflow Development Tutorial

Can be found in Chapter 5. Designed to give a short introduction into the principles of developing integration solutions with the Connecting Software Connect Bridge.

The CB SQL Reference

In Chapter 6, a reference of the CB variation of the SQL Language used to control the CB Server is provided. You will find that it closely matches ANSI-SQL.

1.2 Operating Principles

The Connect Bridge is a powerful, yet simple-to-use solution to connect standard servers like Microsoft Exchange, SharePoint or Dynamics CRM either with each other, or with third- party products.

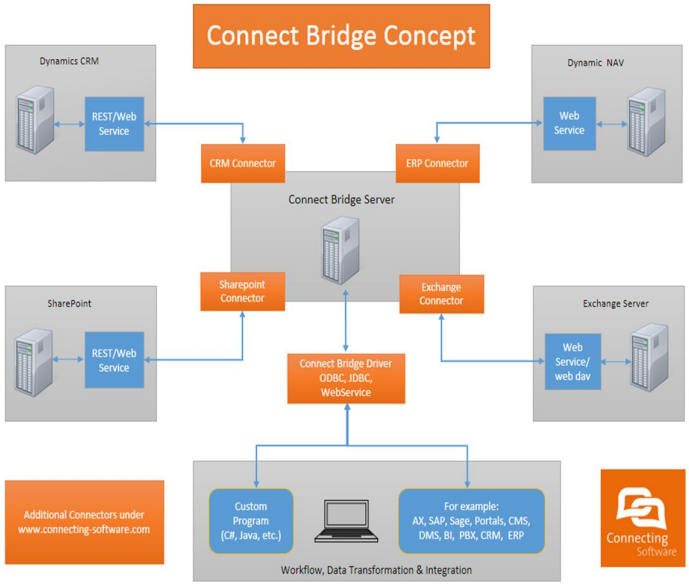


Figure 1 - Server-to-Server

The principle is simple. As depicted in Figure 1 - Server-to-Server, the CB Server connects to standard servers with a proprietary Connector that uses standard interfaces like WebServices, REST, etc. to access entities of the destination server. The data integration is maintained this way. A custom program then accesses the CB Server either via CB ODBC, JDBC or Webservices driver and uses CB SQL to control the integration. "SELECT FROM Appointments" or "INSERT INTO Contacts" can then be used instead of having to learn the intricacies of the individual interface.

In the second scenario (as depicted in Figure 2 - CB Client-Server Scenario), software vendors of specialised LOB[1] or Portal solutions can use the CB to access functionality of Standard Servers quickly, safely and effectively. A typical example would be to access Exchange accounts from within the Portal or obtain CRM contacts from within an industry specific LOB.

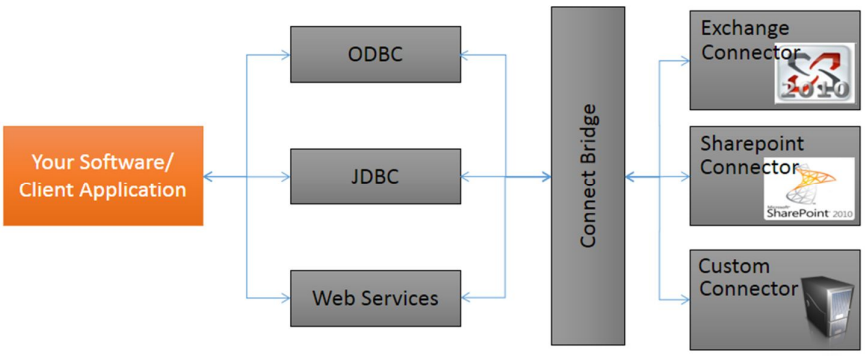


Figure 2 - CB Client-Server Scenario

For more details on how to connect to the CB Server with custom applications, please check the Workflow Tutorial in this manual and visit the Developer's Corner on our webpage www.connecting-software.com.

2 CB Administration Tool

The CB Server Administration Tool provides functionalities to manage and configure the Connect Bridge. The tool provides a friendly Graphical User Interface (GUI) capable of managing and configuring the entire CB-Server installed on a local PC or on a remote server. It uses simple query statements and specific System Stored Procedures to provide all management features in the Connect Bridge.

It is a client tool which communicates via CB-ODBC Driver with the internal CB Administration Connector of the CB-Server. The CB Administration Connector is the CB module responsible to communicate with the internal database system, therefore providing all mechanisms to interact with it, including security, performance and scalability.

2.1 "Connect To" CB-Server

Start the CB Administration Tool by clicking the "Windows" (Start) button, then go to "All Programs", choose the folder "Connect Bridge", then choose "Connect Bridge Server Administration Tool".

After startup, the following dialog appears (see Figure 3 - Admin Tool: Login Screen). You need to provide the host name (or IP address), where your CB-Server is installed (Server), the User name, its Password and the Port for communication via CB-ODBC Driver (the default is 8087).

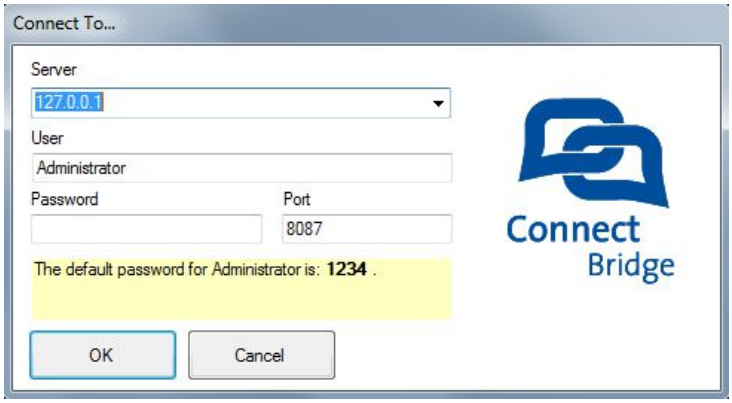


Figure 3 - Admin Tool: Login Screen

Hint: in this case, the CB-Server was installed on the local machine, therefore the Server entry would be localhost *or its IP address (127.0.0.1). The* Administrator *user is created automatically during CB installation in the system. The initial password is shown in the note bellow the password box. Upon first login, you will be prompted to change the password (see* Figure 4 - Change Password Dialog*).*

Figure 4 - Change Password Dialog

It is highly recommended to change the password of the Administrator **user after the first login. In order to do that, go to "Manage" then select Users in the left pane, then click on the Administrator User and click Edit. The dialog screen displayed allows you to change the password. Then click Save.**

Once the connection to CB-Server is established, main window of CB Server Administration Tool is shown (see Figure 5 - CB Admin Tool: Main Window).

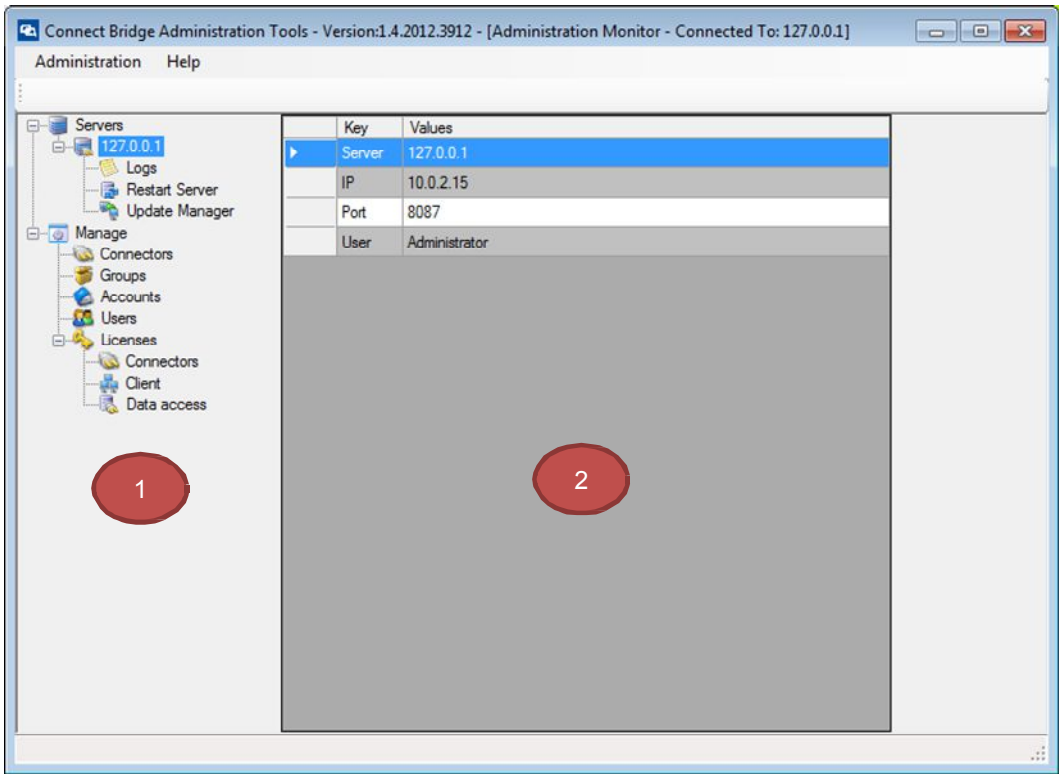


Figure 5 - CB Admin Tool: Main Window

2.2 Description of the User Interface Components

Navigation Panel (1) – refers to the panel on the left side of the main window. It contains 2 selection trees: Serversand Manage.

Grid Editor (2) – the grid (table) in the middle, shows the names (the column Key) and values (the column Values) of the properties selected in the Navigation Pane.

Items in Navigation Panel:

Servers– The list of actions you can execute against a connected CB-Server. By the time of this documentation release, the client is only able to connect to one CB- Server at a time.

-> Logs– Allows the management of the logging settings of the CB-Server.

-> Restart Server– Restarts the CB-Server the tool is connected to.

-> Update Manager– *For internal purposes only.*

Manage– the list of entities you can manage via CB Server Administration Tool.

-> Connectors – Allows you to manage (install, modify, uninstall) CB-Connectors, which are installed on the CB-Server.

-> Groups– Allows you to manage groups of users and their access to accounts

-> Accounts – Allows you to manage CB-Accounts that access servers via connectors

-> Users– Allows you to manage Users

-> Licenses– Allows you to manage Licenses

-> Data Providers– Allows you to manage Data Providers (local or remote).

Description of the Menu Items:

Administration

-> Connect To – Allows you to establish a new connection to a CB-Server (local or remote).

-> FTP Manager – Offers the possibility to transfer files via ftp to the CB-Server from a client machine connected to the CB-Server.

Help

-> About Connect Bridge Administration Tool – Provides information about the Connect Bridge Administration Tool and CB- Connectors.

-> Request new license – Allows you to request a new Connecting Software Connect Bridge license from Connecting Software.

-> Activate license – Allows you to activate the Connect Bridge license provided by Connecting Software.

2.3 Managing Logs

You can choose the output location of the log files from the Navigation Panel - expand Servers, then expand a server you are connected to (e.g. localhost) and click on Logs. Figure 6 - Log Dialogs depicts the two screens available within the Log Dialog.

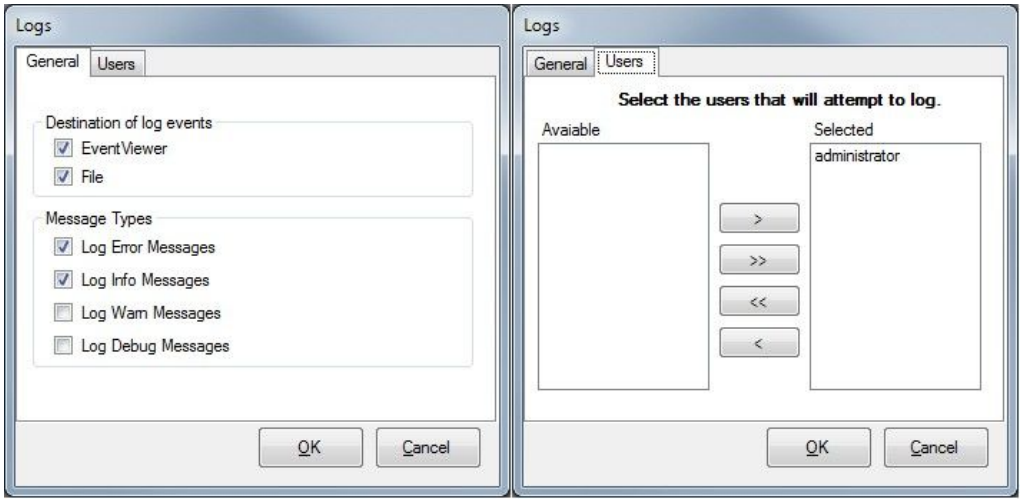


Figure 6 - Log Dialogs

On the tab Generalthe group Destination of log events allows to select wherethe logged events should be saved to. Options include either the Windows Event Viewer or a physical file that can be found in the MgServerService folder of the CB installation(usually "*C:\Program Files\Connecting Software\Connect Bridge\CbServerService\logOutputfile.txt*".

The group Message Types allows to specify what kind of messages should be logged:

Log Error Messages– CB errors will be logged.

Log Info Messages– All information messages will be logged.

Log Warn Messages– All warning messages will be logged.

Log Debug Messages – Full debugging info from the CB-Server will be logged. This choice produces the most detailed output, but also represents the biggest demand on the computer resources.

The Users tab allows you to decide for which users these logs should be stored. All potential users are shown in the list Available. By selecting one or more and clicking on the Arrow Right Button, the respective user is chosen and therefore included in the logging. During the installation of the CB, user Administrator is created and automatically moved to the list box Selected. Thus, his actions on the server will always be logged.

When finished, click on the OK button. You are asked to restart the Connecting Software Connect Bridge-Service in order to apply the new settings. From that point on, the logging behaves according to the new configuration.

Warning! Extensive logging can create significant amounts of log files. It can also be a substantial drain on the CB server resources, thus slowing it down. Connecting Software therefore recommends to use minimal logging during normal operations and extended logging only when trying to find errors.

2.4 Managing Connectors

As described in the introduction, the CB Server uses CB Connectors to connect to third- party products like MS Exchange or MS SharePoint. This enables users of the CB to access the full functionality of the respective server without having to cope with the intricacies of each individual interface.

A significant amount of CB-Connectors is already available, including but not limited to:

2.4.1 MS Exchange

2.4.2 MS SharePoint

2.4.3 MS Dynamics CRM

2.4.4 MS Dynamics NAV

2.4.5 etc.

Connecting Software and partners are constantly working to expand the number of available connectors. Please check our webpage www.connecting-software.com or contact a Connecting Software representative or partner to get an updated version of the CB- Connector list.

Connectors can be installed, managed and uninstalled in the Connector section of the CB Admin Tool. Open the Admin Tool, navigate on the left pane down to Manage and Connectors then double click Connectors (see Figure 7 - Connector Administration).

The Middle Pane (Grid) will display a list of installed connectors. Description of columns displayed in the grid:

ID– CB internal ID

PluginEnabled –if checked, the connector is enabled, if unchecked it will be disabled.
PluginName – Name of the CB-Connector (important for use in the Accounts section)
PluginPath– Designates the installation path of the Connector.

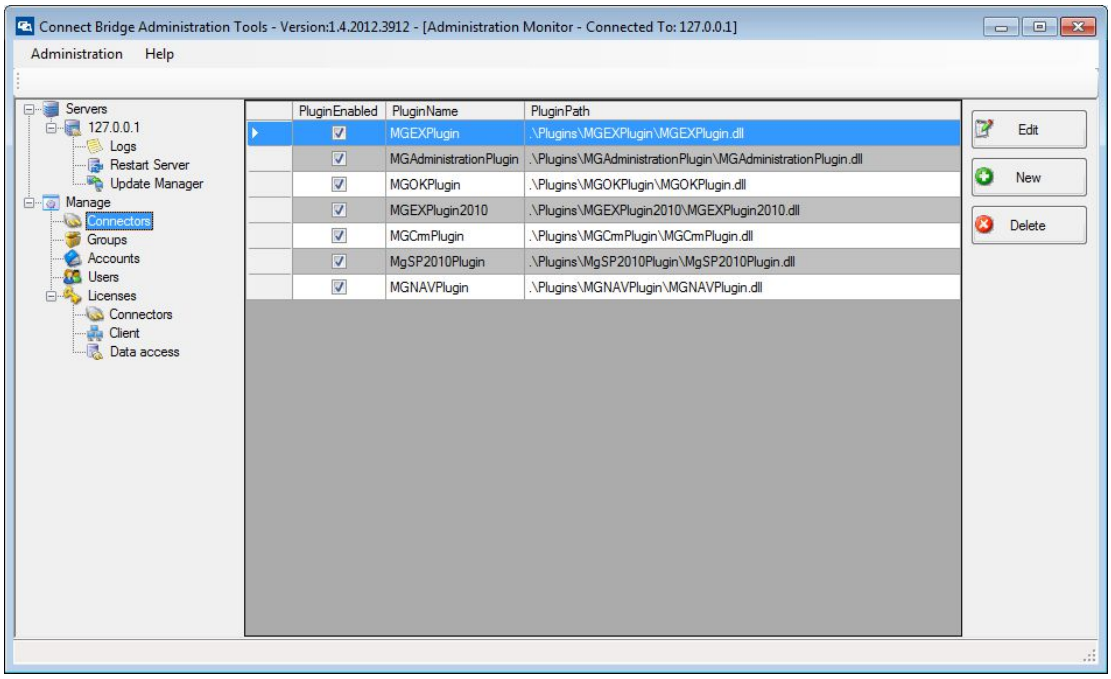


Figure 7 - Connector Administration

The following actions can be performed on/with a connector:

1) Enable/Disable a CB-Connector

Select a connector in Grid and click on Edit button on the right panel in the main window. You cannot change any settings of property for the installed connector. These settings are part of the connector library configuration and are pre-configured settings from the installation. The only available option is to enable/disable the Connector by ticking the checkbox next to the name.

2) Install a new CB-Connector

The standard CB installation comes with a set of connectors. Should you have purchased additional connectors, you need to add them first and then request a license.

When you receive a Connector, please unzip it to the Plugins directory of your server installation (usually: *C:\Program Files\Connecting Software\Connect Bridge\CbServerService\Plugins*) – see below

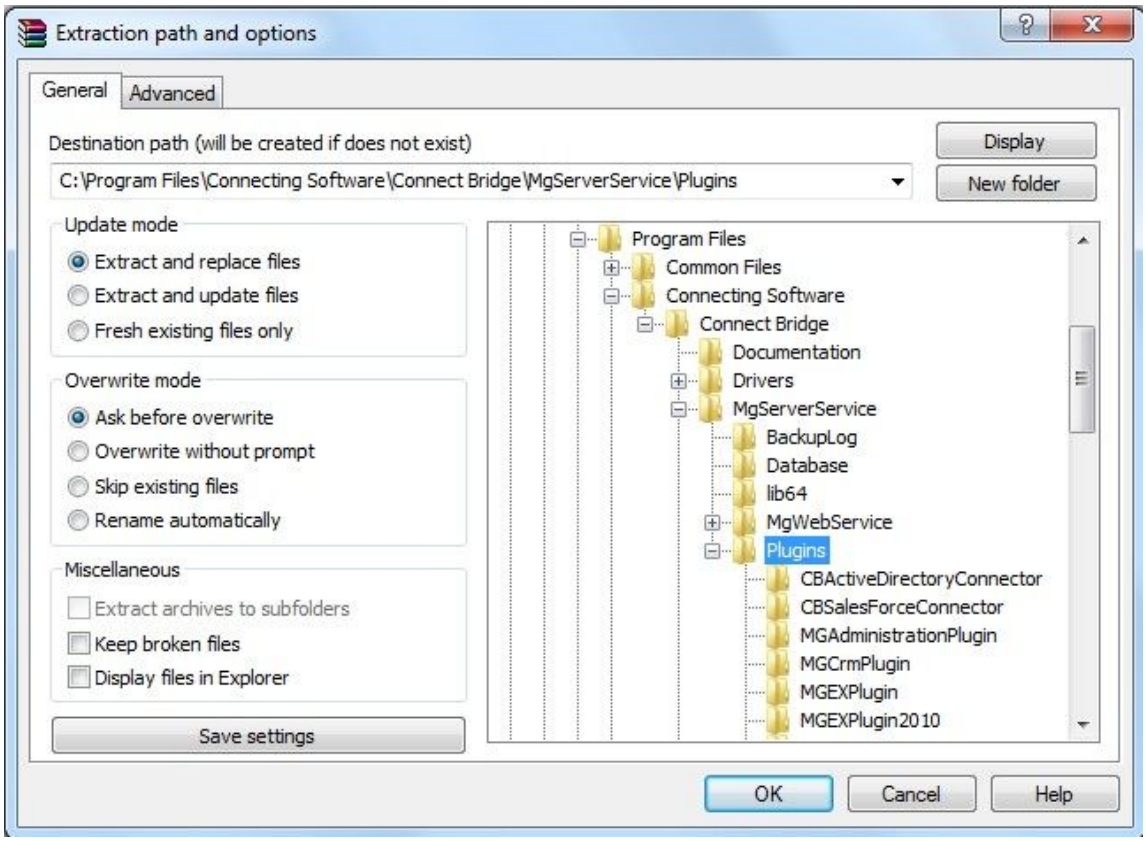


Figure 8 - Unzip Example

Once unzipped, open the Admin Tool, double click on Connectors [1], then New [2], give it the same name as the file you have received without any extensions (e.g. MGWNAVPlugin) [3]. Click the ellipsis button (…) [4], browse to the Plugins Directory and select the appropriate dll file (e.g. MGWNAVPlugin.dll) [5]. Then click Save[6]

–see Figure 9 - Connector Installation.

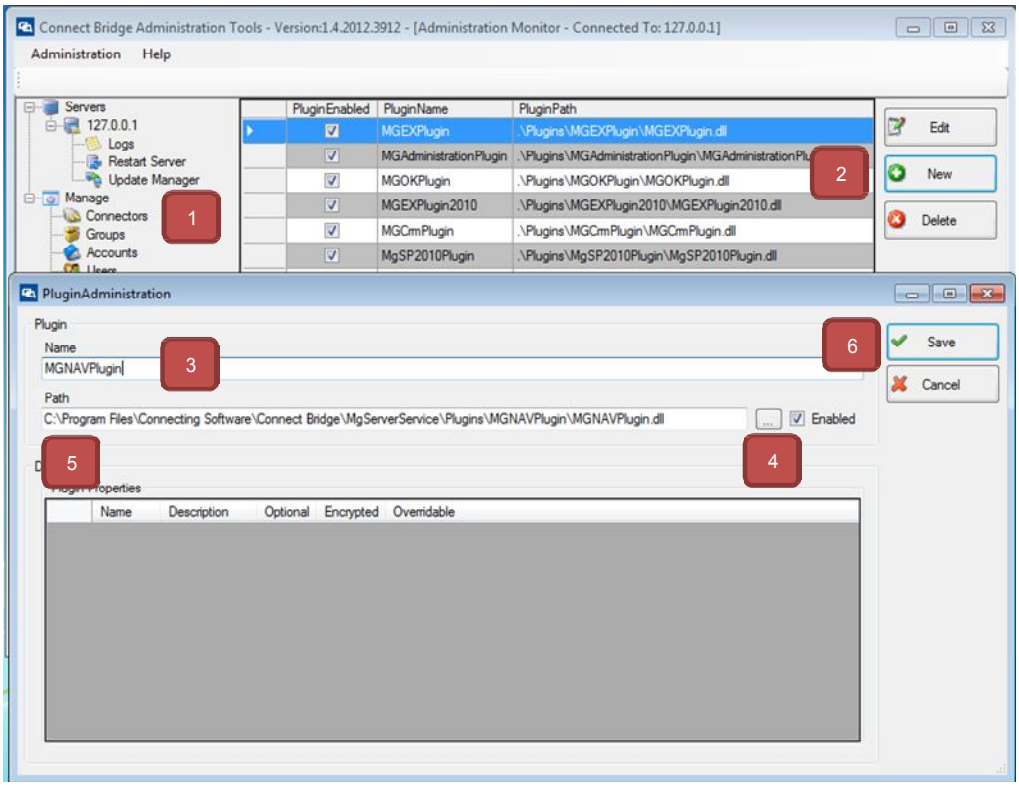


Figure 9 - Connector Installation

Now you need to request a license for the newly installed Connector. Please make sure that you describe in the message body which Connector exactly you want the license for.

3) Uninstall a Connector

In order to uninstall a Connector, select the Connector in the middle pane (grid), then click

Delete

Delete does not remove the Connector from the computer; it merely makes it unavailable to the CB. If you want to delete it completely, delete it from the Plugins installation folder (usually *C:\Program Files\Connecting Software\Connect Bridge\CbServerService\Plugins\).* Should you accidentally delete a Connector, you can add it again (same procedure as installation), but you need to reconfigure the accounts as they will have lost the reference.

2.5 Managing Accounts

(CB) Accounts are used to access a target server like MS Exchange or MS SharePoint via the CB Connector. These can be actual User Accounts like "John Doe" or Service Accounts like "ServiceCB". The actual user Accounts are usually chosen in case of a client- server scenario (e.g. a proprietary portal or ERP system accesses MS Exchange Emails for each user via the Connect Bridge). Service Accounts are usually chosen for server- server integrations (e.g. synchronizing appointments between Dynamics CRM and MS SharePoint).

In either case, the Account needs to exist in the domain first, then registered with the appropriate rights in the target server. After that, the account can be created and configured in the CB Admin Console (for details, please check the CB Installation & Configuration Guide).

Accounts can be managed in the Accounts section of the CB Admin Tool. Open the Admin Tool (see Figure 10 - Account Administration), navigate on the left pane [1] down to Manage/Accounts then double click on Accounts. The middle pane (Grid) [2] will show you all the existing accounts[3].

Description of columns displayed in the grid:

2.4.6 ID– CB internal ID of the record

2.4.7 AccountName– represents the name of the account.

Tip: *Use the prefix "*`ACCOUNT_`*" or "ACC_" in order to facilitate recognition of the* `Account Name` *inside a connection string.*

By default, CB creates one account per installed Connector

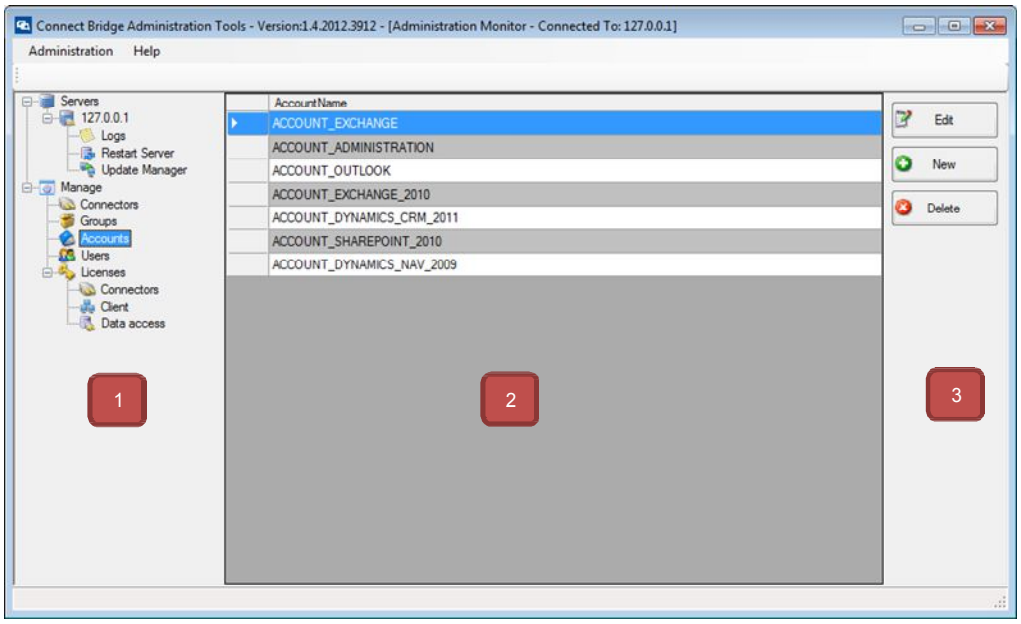


Figure 10 - Account Administration

The following actions can be performed on/with an account:

a) Create a new account.

Click on New button on the right pane [3]. A dialog window appears (see Figure 11 - Account Management):

1. Account Name– The name of the account
2. Connector – Select the name of the Connector, you want to use with the account. One account can only work with one Connector – please remember that when naming the accounts
3. Connector Properties – Fill the values of required properties for corresponding Connector (this is connector specific – see the respective CB Connector Reference and check the CB Installation and Configuration Guide for the most common connectors).
4. When you have finished the configuration, click the Save button in order to apply these settings.

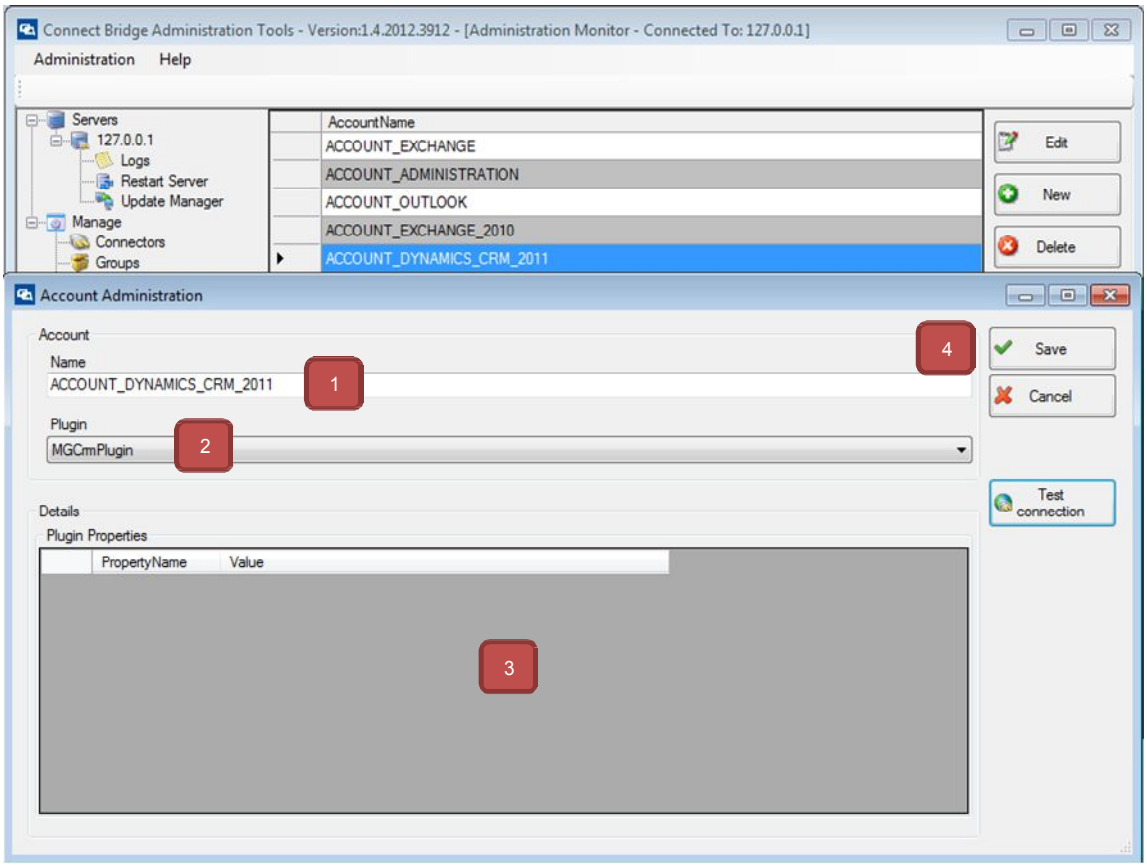


Figure 11 - Account Management

b) Edit an exisiting account

The only difference to a new account is that you have to select the account in the middle pane and click Edit. Apart from that, the same steps apply as described in a) Create a new account

c) Delete an exisiting account

In order to deltete an account, select the account in the middle pane and click Delete.

Warning! This will terminate access to the server connected via that particular account for all groups that use this account and subsequently to all users assigned to the respective group!

2.6 Managing Groups

(CB) Groups are organizational units that not only help organize users into functional groups, but also provide the link to Accounts.

That means a Group has permissions for one or more Accounts and Users belong to one or more Groups.

Groups can be managed in the Groups section of the CB Admin Tool. Open the Admin Tool (see Figure 12 - Group Administration), navigate on the left pane [1] down to Manage and Groups, then double click on Groups. The middle pane (Grid) [2] will show you all existing Groups[4].

Description of columns displayed in the grid:

2.6.1 ID– CB internal ID of the record

2.6.2 GroupName– Represents the name of the group

Tip: *Use the prefix "group_" or "GRP_" in order to facilitate differentiation between Users and Groups*

By default, CB creates a standard group called group_administrators.

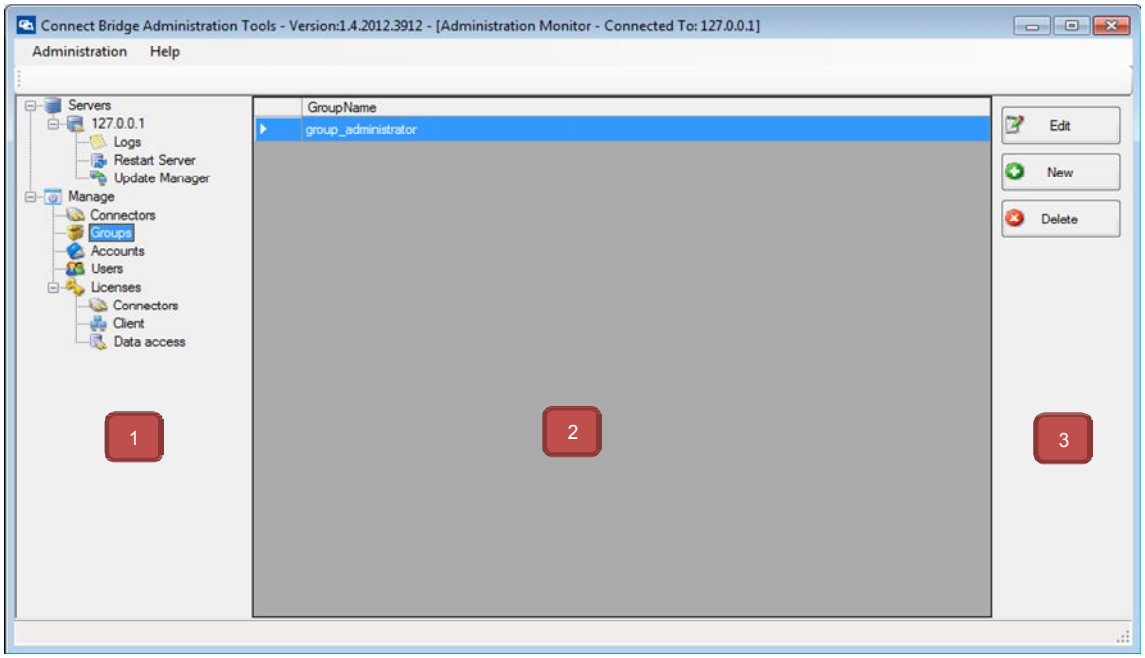


Figure 12 - Group Administration

The following actions can be performed on/with a Group:

a) Create a new group.

Click on New button on the right pane [3]. A dialog window appears (see Figure 13 - Group Management):

1. Group Name– The name of the group (for naming see the tip above)
2. Select the accounts you wish to give this group access to by clicking on them in the Account Detail Window
3. Click on the *Arrow Right Button* to add them to the list. If you wish to add all accounts, there is no need to select them all. Just use the *Double Arrow Right Button. Arrow Left Button* removes a single account, *Double Arrow Left*removes all

Arrow	Name
>>	*Double Arrow Right Button*
>	*Arrow Right Button*
<	*Arrow Left Button*
<<	*Double Arrow Left Button*

Table 1 - Group Selection Arrows

4. When you have finished the configuration, click the Save button in order to apply these settings.

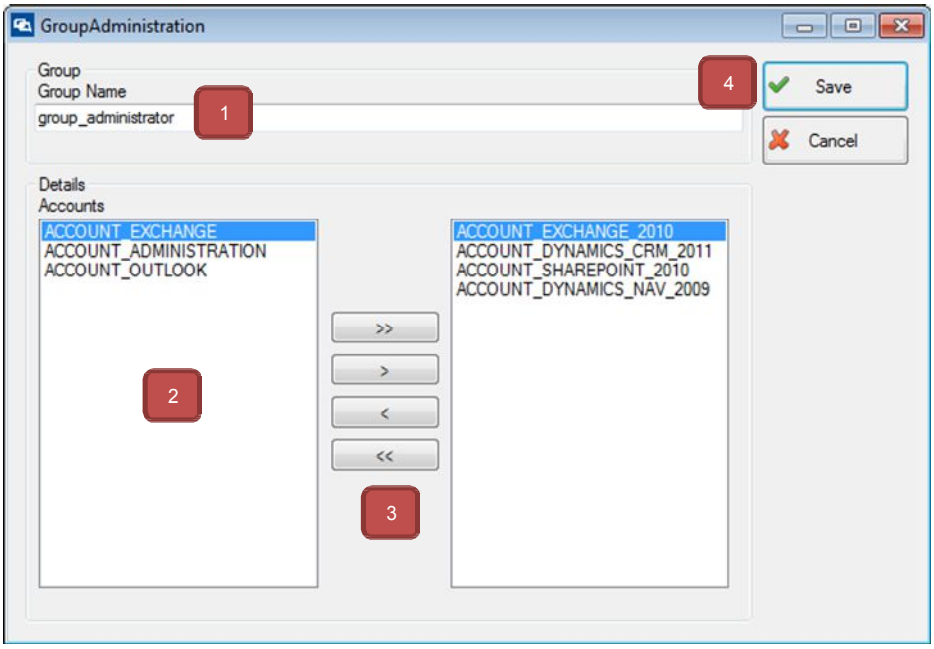


Figure 13 - Group Management

b) Edit a group

You can change which accounts the group has access to, by selecting the Group in the Middle Pane of the Group Administration Window (see Figure 12 - Group Administration [2]), then click Edit. The remaining steps are identical to the ones described in a) create a new group

c) Delete a group

You can delete a group by selecting the Group in the Middle Pane of the Group Administration Window (see Figure 12 - Group Administration [2]), then clicking Delete. **Warning! This will cancel the permissions of any user that is a member of that particular group for the accounts assigned to the group!**

2.7 Managing Users

(CB) Users are either domain users that access Connect Bridge functionality via their host application in client-server mode (e.g. a portal or an industry-specific ERP solution), or they are Service Users that represent the access rights of a particular application or developer (e.g. a program that handles the synchronization workflow of contacts between CRM and NAV).

Quite often, users might ask why there is a separation of Users and Accounts, as these follow a similar logic. The reason is simple – the User represents the person or service that "consumes" CB services, while the account represents the access to a target server via a CB Connector. While there are cases, where users and accounts will be identical (e.g. when a user accesses his exchange mailbox via the Connect Bridge from his ERP system), there are many cases where solution providers want to differentiate between the user that accesses the CB and the account that accesses the target server (e.g. when a teamsite owner in SP starts a synchronization workflow with CRM where it has no access rights).

In order to allow for maximum flexibility in the implementation of such integration solutions, Connecting Software has decided to differentiate between users and accounts.

Tip: *If you need help to implement a security and role model that complies with your company's compliance rules, please contact Connecting Software or a certified integration partner for consulting support*

Users can be managed in the Users section of the CB Admin Tool. Open the Admin Tool (see Figure 14 - Users Administration), navigate on the left pane [1] down to Manage and Users then double click on Users. The middle pane (Grid) [2] will show you all existing Groups[5]

Description of columns displayed in the grid:

2.6.3 ID– CB internal ID of the record

2.6.4 UserLogin– The login name of the user (usually the domain name)

2.6.5 UserName – The name that is used in the connection string (can be identical with UserLogin)

By default, CB creates a standard user called administrator, this user is automatically assigned to the default group_administrators.

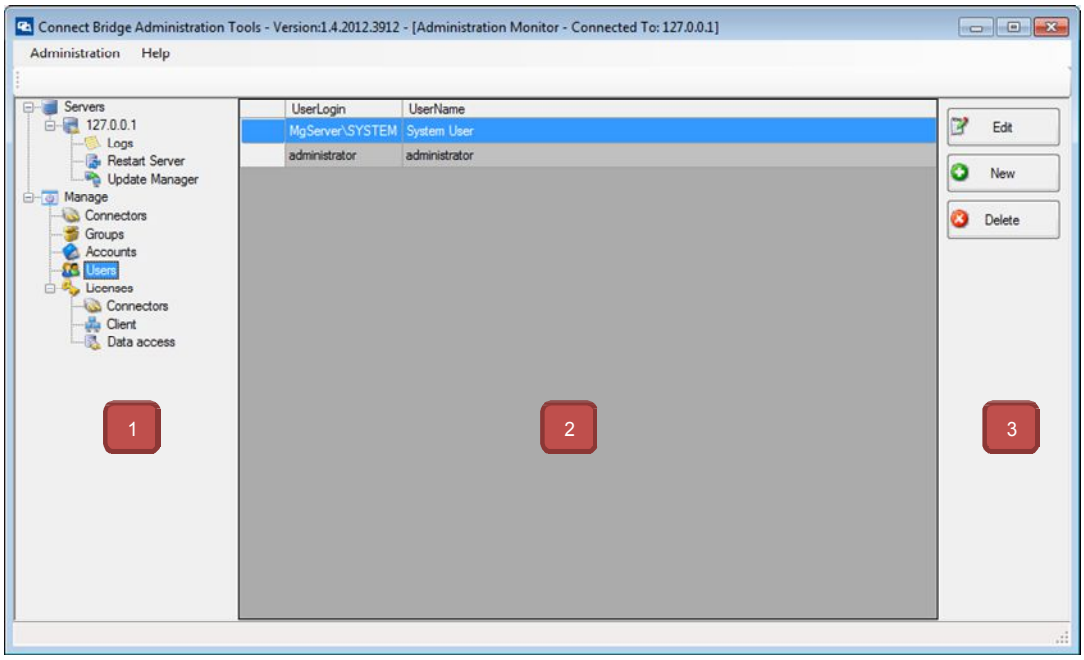


Figure 14 - Users Administration

The following actions can be performed on/with a User:

a) Create a new user

1. Open the Admin Tool and double click on Users
2. Click on New button on the right pane [3]. A dialog window appears (see Figure 15

- Users Management)

3. Provide a meaningful name (one that is used within the CB, the user login[6] and password and retype the password
4. Select "*User and Password"* if you do not want to use a Domain User or "*Windows Authentication*"[7] if you want to use a Domain User and Windows Authentication
5. Select the groups you wish to give this user access to by clicking on them in the Account Detail Window, then click on the *Arrow Right Button* to add them to the list. If you wish to add all users, there is no need to select them all. Just use the *Double Arrow Right Button. Arrow Left Button* removes a single user, *Double Arrow Left* removes all (for a description of the arrows, please see Table 1 - Group Selection Arrows).
6. Click on Save

In case of User Authentication, these can be identical. In case of Windows Authentication, however, it might be practical to use a shorter User Name to avoid having to use the fully qualified domain name every time.

If *Windows Authentication* is greyed out, you haven't used a properly qualified domain name like *mydomain\username* in the field User Login

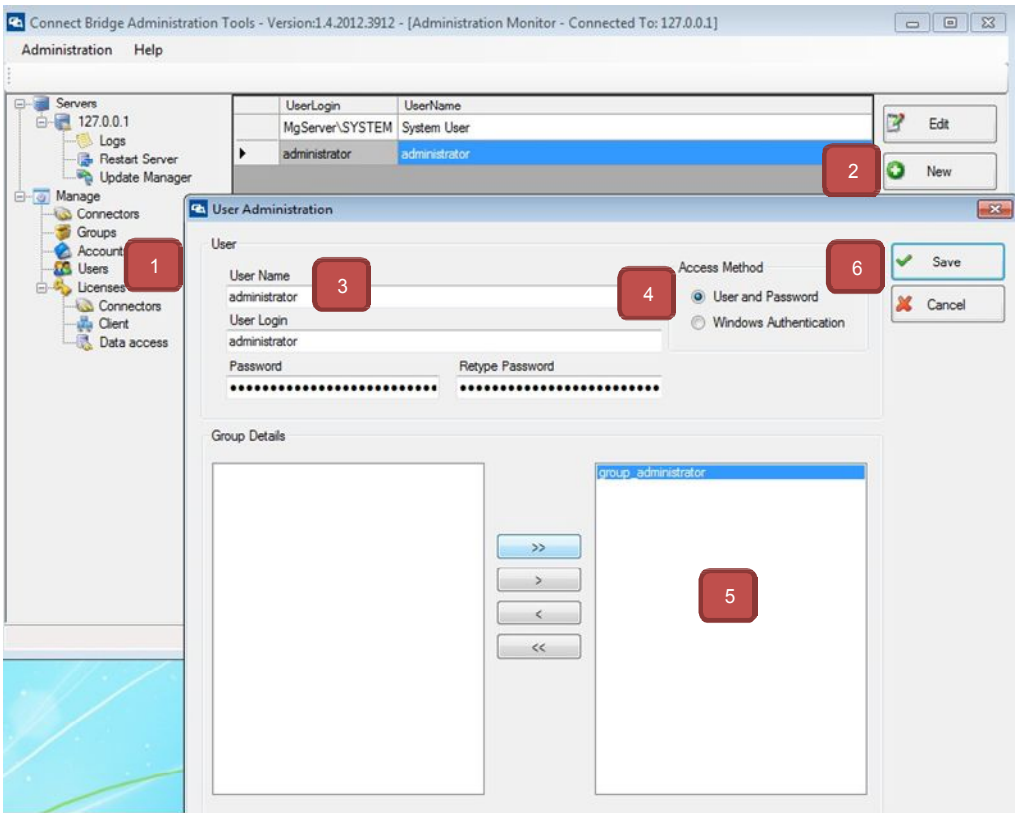


Figure 15 - Users Management

b) Edit an existing User

By selecting an existing user in the middle pane (grid) [2] of the Users Administration Window (see Figure 14 - Users Administration) and then clicking Edit changes in the assignment of the user to a particular group or password changes can be done. To change a password, simply type a new one in the field Password and type it again for confirmation purposes in the Retype Password field, then click Save. To change the assignments use the arrow key as described in Table 1 - Group Selection Arrows.

c) Delete an existing User

In order to delete an existing user, select the user in the middle pane (grid) [2] of the Users Administration Window (see Figure 14 - Users Administration) and then click Delete.

Warning! Once a user has been deleted, all connection strings that use the particular user will no longer work! Please make sure you contact the people responsible for integrations applications and make sure that they are aware of the deletion before you actually delete.

3 The CB Service Controller

A neat little tool that should help you to quickly resolve common administration issues like starting and stopping (and checking the status of) the server service, requesting and activating a license or looking at log files.

It can be found in the Connect Bridge Folder of the Start Menu

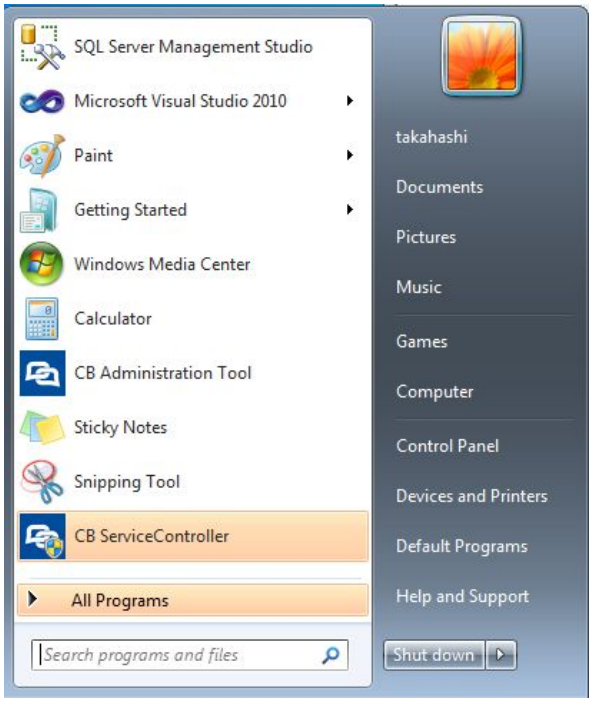


Figure 16 – Connect Bridge Folder in Windows Start Menu

Once started, it will appear in the System Tray on the lower right hand side where either a double-click (which will show the dialog depicted in Figure 18 - CB Service Controller Window) or a right-click on the Connecting Software Icon will give you access to the provided functionality (see Figure 17 - CB Service Controller Icon). If you cannot find the Service Controller, you did not install the server components of the CB. Please install or connect remotely to the machine where they are installed.

In case the Complete installation option was chosen, the service should run automatically.

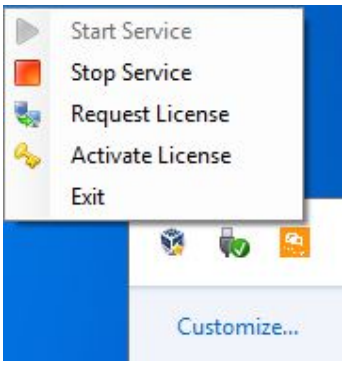


Figure 17 - CB Service Controller Icon

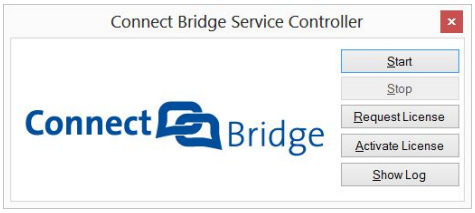


Figure 18 - CB Service Controller Window

The features provided by the CB Service Controller are:

- Start button to start the CB Server Service.
- Stop button.to stop the CB Server Service
- Request License – By clicking on this button you can send a license request to Connecting Software. You need to define the location where the license file has to be stored in your computer. In case you do not have a default mail application configured, you will receive a message indicating that you will have to send the request manually via your preferred mail client to the Connecting Software support team.
  - Activate License – By pressing this button you will be asked to provide the location of the license file. The file is generated by Connecting Software based on your request. After validation, you can activate the license in your software once you have received it from Connecting Software via Email.
  - Show Log – By clicking on this button, a list with your activity logs will be shown (server side logs only).

4 The CB Query Analyzer

The Connect Bridge Query Analyzer (QA) is a client tool designed to access the CB Server. It can be used to run CB SQL statements against all target servers connected to the CB server via CB Connectors.

The main purpose of the tool is to test out CB SQL statements before embedding them into proprietary workflow and data transformation code. It can be also used to train CB SQL and to get to know the entity model of the connected servers.

Technically, QA operates by parsing the SQL statement or Stored Procedure, passes it on to the CB Server via ODBC connection, waits for the response and displays the result.

***Tip:** Please make sure you run the CB Query Analyzer with Administrator rights on your local machine. In order to do that, right-click on the Query Analyzer icon, click on Properties, go to the Tab Shortcuts, click on Advanced and then click the checkbox "Run as administrator"*

4.1 Layout of the CB Query Analyzer

Developers familiar with common database administration tools will find the layout of the QA quite familiar. Connecting Software has strived to make working with QA as easy and straightforward as possible.

The following items are shown in the default view:

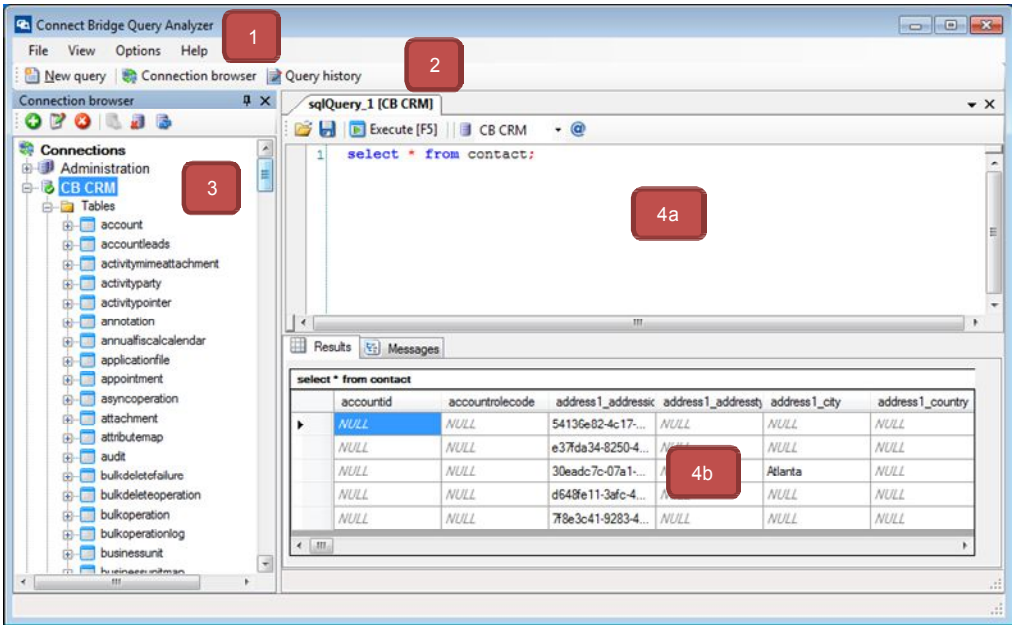


Figure 19 - Query Analyzer: Standard View

The Menu Bar [1] – a standard menu bar contains the following items:

Level 1	Level 2	Level 3	Description
File	New	Query	To generate a new query against a connection
		Connection	To create a new connection to a server
	Open	Script File	Opens a text file that Contains pre-generated SQLstatements
	Close		Closes the application
View	Connection browser		Shows and activates the Connection browser
	Event viewer		Shows and activates the Event viewer
Options	Settings		Displays settings for logging options
Help	About		Shows the About window

The Icon Bar [2] – contains icons for common tasks

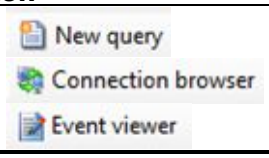
Icon	*Description*
	Opens the Query window for the selected connection Opens and activates the Connection browser Opens and activates the Event viewer

Table 2 - Icon bar icons

The Connection Browser [3] on the left – showing connections to both the CB server (shown as Administration), as well as connections via the CB server to target servers like MS SharePoint and MS Exchange.

The default view of the QA only shows one connection, the one to the CB Server (called Administration). Later, other connections can be added.

The green Plus button allows to create a new connection (alternatively, click on the Menu File | New | Connection. That will open the New Connection dialog.

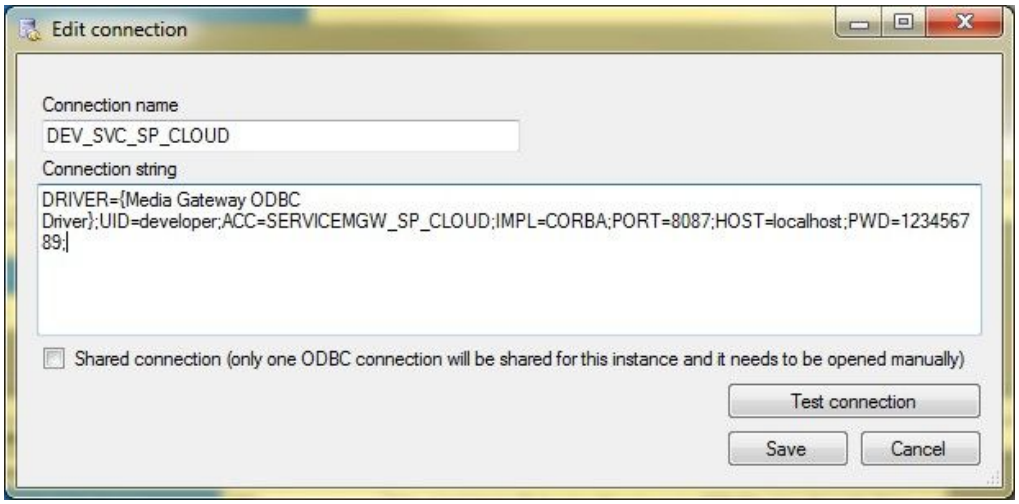


Figure 20 - New/Edit Connection Dialog

Give the new connection a meaningful name[8] , add the appropriate connection string, optionally click on Test connectionand click Save

By clicking on the + Button next to the name of the connection, the tree is expanded, a connection established and the user can take a more in depth look at Tables (i.e. Entities) and Stored Procedures available.

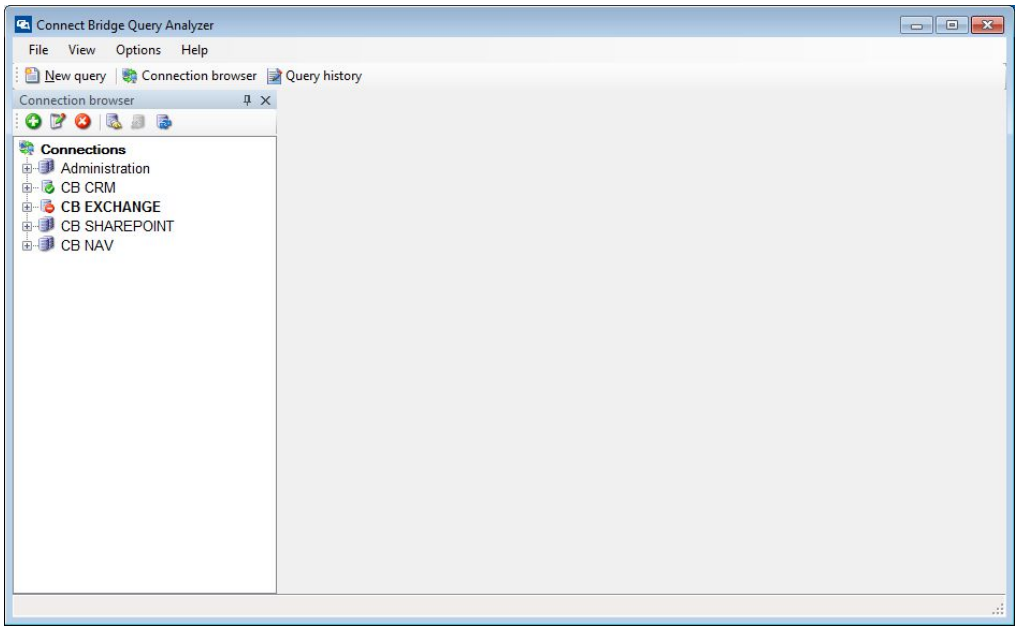


Figure 21 - Connection Browser: Tree Expanded

If the CB was able to establish a connection, a green tick mark is displayed next to the name of the connection on top of the database icon (see CB CRM in Figure 21 - Connection Browser: Tree Expanded). If the connection fails, a red "Do not enter" sign is displayed (see CB EXCHANGE in Figure 21 - Connection Browser: Tree Expanded).

In that case, right-click on the connection and click on Show Error

Connecting Software recommends to use the following syntax in order to name a connection: USR_ACC_SRV_DOMAIN. E.g. DEV_SVC_SP_CLOUD would be a developer user, using the standard CB service Account connecting to a SharePoint server in the CLOUD (Office365).

In case you cannot find a solution for the error yourself, please check the support page of our website www.connecting-software.com or contact your support representative.

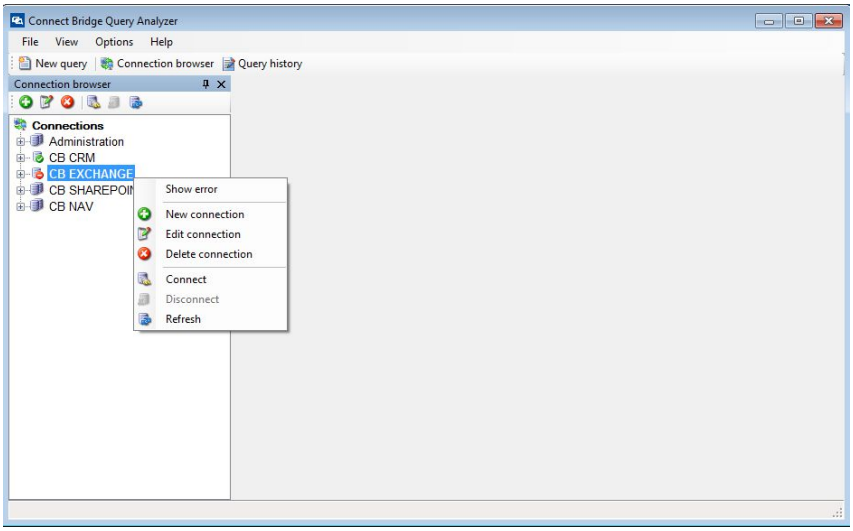


Figure 22 - Connection Error

The following commands are available in the Connection browser (the same commands are also available through the context menu via right-click on the connection – see Figure 22 - Connection Error) – see Figure 23 - Icons: Connection browser

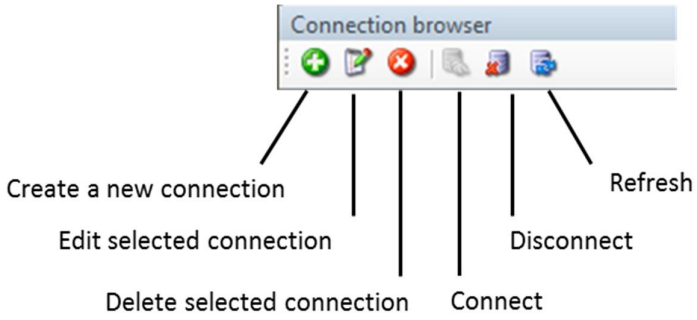


Figure 23 - Icons: Connection browser

Icon	***Description***
Create a new connection	Opens the dialog window to add a connection string to create new connection to a server
Edit selected connection	Opens the dialog window of the selected connections and allows modification of the connection string
Delete selected	Deletes the selected connection
Connect	Connects to the Connector/server designated in the connection string
Disconnect	Disconnects the exiting connection
Refresh	Refreshes the existing connection

Table 3 - Connection browser icons

Expanding the Tables section will show the entire set of entities available to the user, expanding those will show details regarding "Columns" (i.e. Properties). As depicted in Figure 24 - Connection browser: Tables & Columns Announcements, Attachments and Calendar are Entities from the SharePoint, AllDayEvent, CategoryValue,

etc. are properties of the calendar entity (table). The property type can be seen in brackets

- Boolean for AllDayEventand String for CategoryValue.

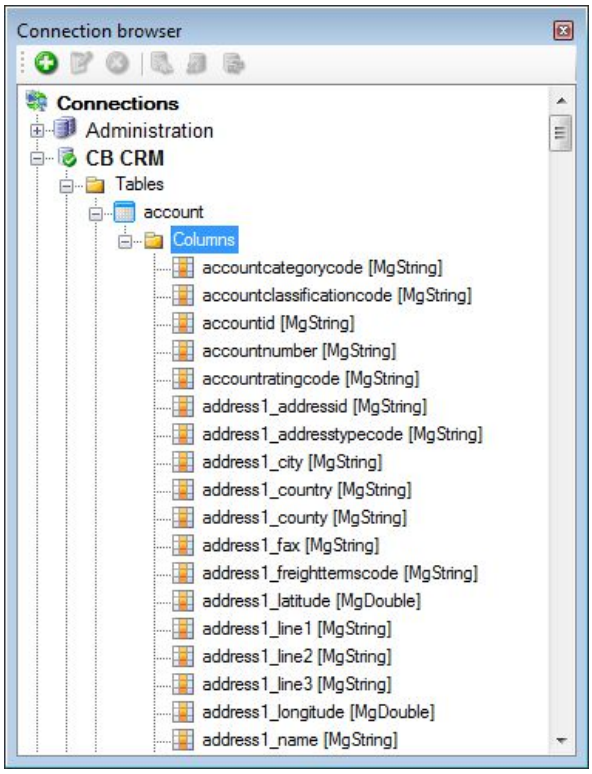


Figure 24 - Connection browser: Tables & Columns

The Query Editor [4a] & [4b] – contains following windows (see also Figure 25 - Query Analyzer: Expanded View)

The SQL statement editing window [4a] – where you write and execute SQL statements

Parameter editing window (extended view [4c]) – where you can define parameters used in SQL Stored Procedures. Click on the blue @ button in the SQL editing window to open.

Results [4c] – shows the result of the executed statement(s)

Messages expanded view [4d] – shows information concerning the last executed SQL statement(s)

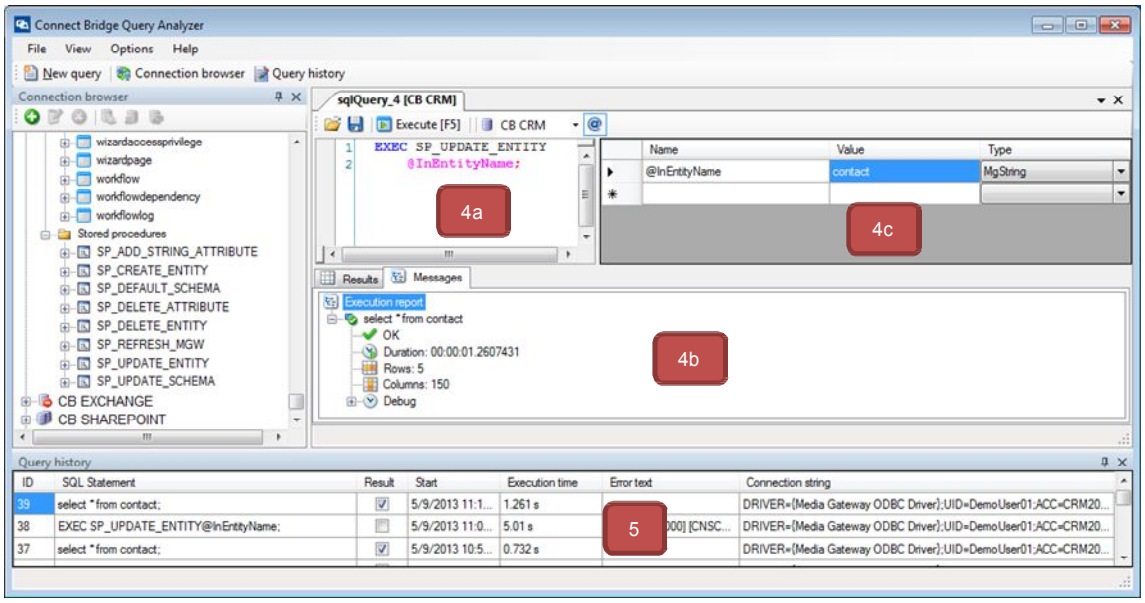


Figure 25 - Query Analyzer: Expanded View

The Event viewer window [5] - shows the history of the executed statements with their properties.

4.2 Working with the CB Query Analyzer

In order to work with the Query Analyzer, the following steps need to be executed:

1. Create one or more new connections
2. Connect to one or more server via the created connections
3. Execute a CB SQL statement or Stored Procedure
4. View Results/Errors in the appropriate Tab in the lower part [4b or 4d] of the Query Editor

To create a new connection to a Connector/server, create on New Connection either by the (green) plus icon or by right-clicking any connection and clicking New Connection in the context menu.

A dialog screen appears that requires entry of a name[10] of the connection string and the connection string itself.

Connecting Software recommends to use names that include Name of User, Name of Account, Name of Connector, location of the server: e.g. use DEV_SRV_SP_MYDOMAIN for a development user that accesses a ServiceCB Account of a SharePoint in the local domain called "MYDOMAIN" or use ADM_JD_EX_CLOUD for an administrative user that accesses exchange in the cloud via the John Doe account (if John has previously allowed that).

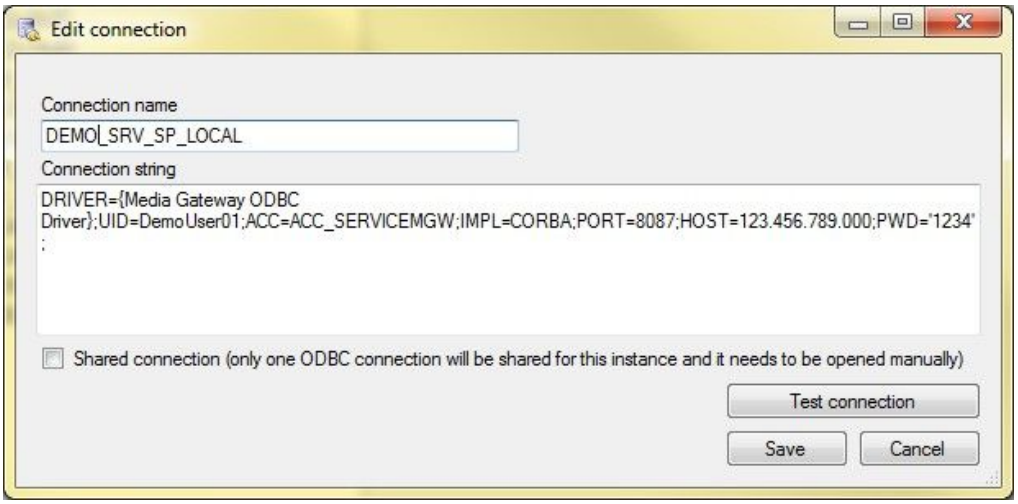


Figure 26 - New Connection Dialog

The Connection string consists of the following parts:

1. DRIVER={Media Gateway ODBC Driver}; This is a fixed part, do not modify.
2. UID=*username*; put a valid user name (e.g. DemoUser01 in the sample shown in Figure 26 - New Connection Dialog) instead of *username*. A user name is valid if it can be seen in the administration tool under the section Users
3. ACC=*nameofaccount*; a valid account name (e.g. ACC_SERVICEMGW in the sample shown in Figure 26 - New Connection Dialog) instead of *nameofaccount*. An account name is valid if it can be seen in the administration tool under the section Accounts.
4. IMPL=CORBA;PORT=8087; is fixed, please do not modify unless you are using a port other then 8087.
5. HOST=*ipaddressORservername*; Please put the name of the server you are connecting to (e.g. SharePoint1) or the IP Address of the target server (e.g. 123.456.789.000in the sample shown in Figure 26 - New Connection Dialog) instead of *ipaddressORservername.*
6. PWD=*'password'*; put the appropriate password for the User instead of *'password'* (e.g. 1234in the sample shown in Figure 26 - New Connection Dialog).

Then choose a connection type. The Query Analyzer can be operated in 2 modes:

Mode	*Description*
Default Mode	Each SQL statement executed will open a new connection to the CB Server. The data is retrieved and the connection closed automatically. This means slower execution than shared mode, but less impact on the licenses used. This mode is recommended for
Shared Connection	Each SQL statement opens a connection and executes the statement, but **doesn't close the connection until the user does it manually**. This mode has a higher performance than the Default Mode, but has a negative impact on the licenses used. Recommended

Table 4 - Connection Modes

Click the checkbox underneath the connection string text box in order to enable Shared Connection, otherwise Default Mode is automatically chosen (see Figure 26 - New Connection Dialog).

Click on the Test Connectionbutton to test the connection. An error message will pop-up, if the connection failed. Please check all parts of the connection string (also in the Admin Tool) as well as any relevant firewall settings!

Click on Saveto save the connection string.

For more details regarding the connection strings, please check the respective Connector Reference.

4.2.1 Connect

To connect to a data sources from QA, go to the Connection Browser window, expand the corresponding connection or select connection name and click on icon Connect in the Connection Browser toolbar above the list of Connections. After the connection has been established, you will see a green checkmark on the left, next to the name of the connection. Should the connection fail, a red & white "Do not enter" icon will appear. In that case, please right-click on the connection and click on Show Errors.

The first time you connect to a new server, the connection will take a bit longer to be established, as the entire schema is downloaded for the first time, so please be patient.

4.2.2 Execute a CB SQL statement or Stored Procedure

Once a connection has been established, click on the connection in the Connection browser and then click on the button New query ([1] in Figure 27 - Query Workflow). Alternatively, simply click on the New query button and choose the appropriate connection from the Connection Window (see Figure 28 - Query Editor Icon Bar)

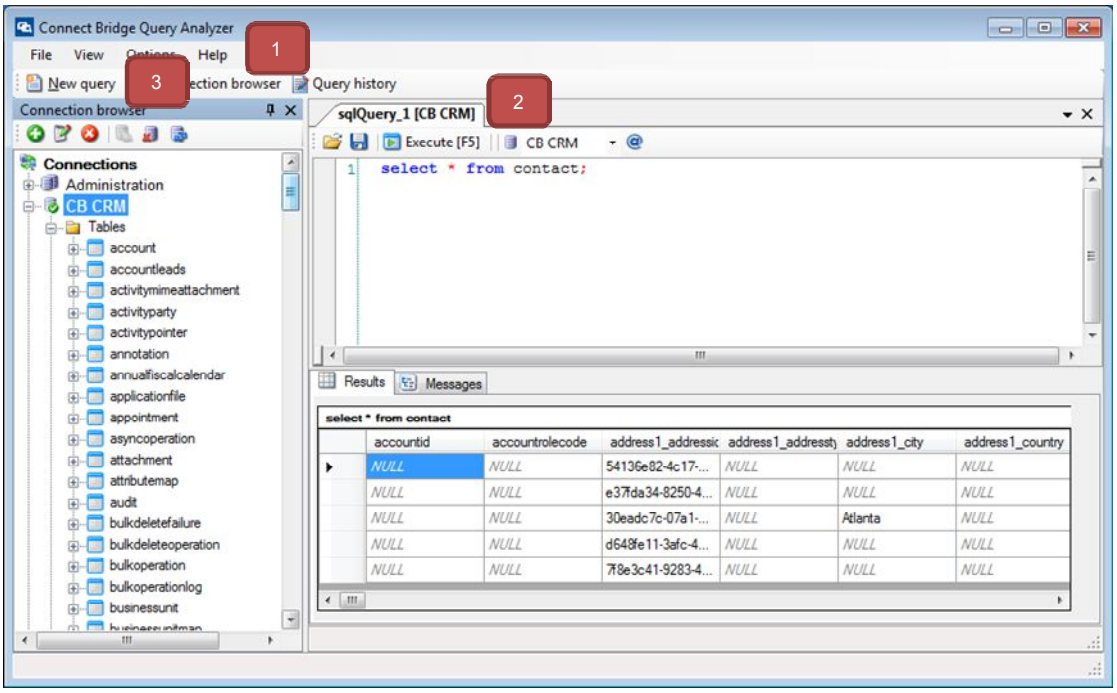


Figure 27 - Query Workflow

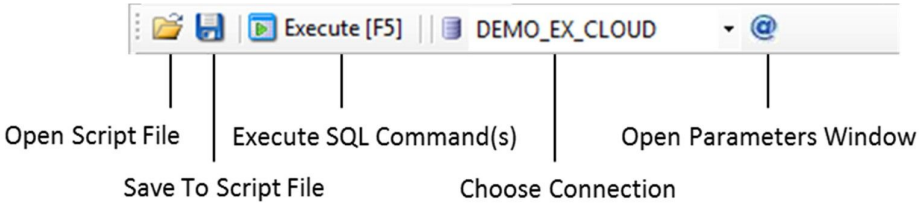


Figure 28 - Query Editor Icon Bar

One Tab [2] will open per New query command. You can alternate between different connections by switching tabs. In the Query Editor Icon Bar (see Figure 28 - Query Editor Icon Bar or [3] in Figure 27 - Query Workflow), choose one or more of the following options:

Icon	***Description***
Open Script File	Opens a script file that contains one or more SQL
Save To Script File	Saves the SQL statements currently displayed in the Query Editor Window [4] to a new script file
Execute SQL Command(s)	Executes the SQL commands currently displayed in the Query Editor Window [4]. Alternatively, press F5 to
Choose Connection	Change the connection the SQL statements currently displayed in the Query Editor Window [4] are executed against.
Open Parameters Window	Opens the Parameters Window (see [4c] of Figure 25 - Query Analyzer: Expanded View) – only needed in case of parameters of stored procedures.

WARNING! It is a common error to choose the wrong connection when executing an SQL statement. So, when you are receiving an Error 42000 for example, you tried to execute a statement against an entity that doesn't exist in the context of the chosen connection (for example Appointment in SharePoint which is called Calendar there). First check if you have selected the correct server in the dropdown menu, then check for spelling errors. That usually solves the problem.

Finally type any SQL statements[11] you wish to execute and press F5 or the Execute SQL Commands button.

You can also auto-generate SQL statements by right-clicking on an entity in the connection browser (for example Contact) and then choosing one of the options in the context menu displayed (see Figure 29 – Auto-generate SQL).

Please check out our SQL Reference in Chapter 6 CB SQL Reference for a detailed description of possible SQL commands.

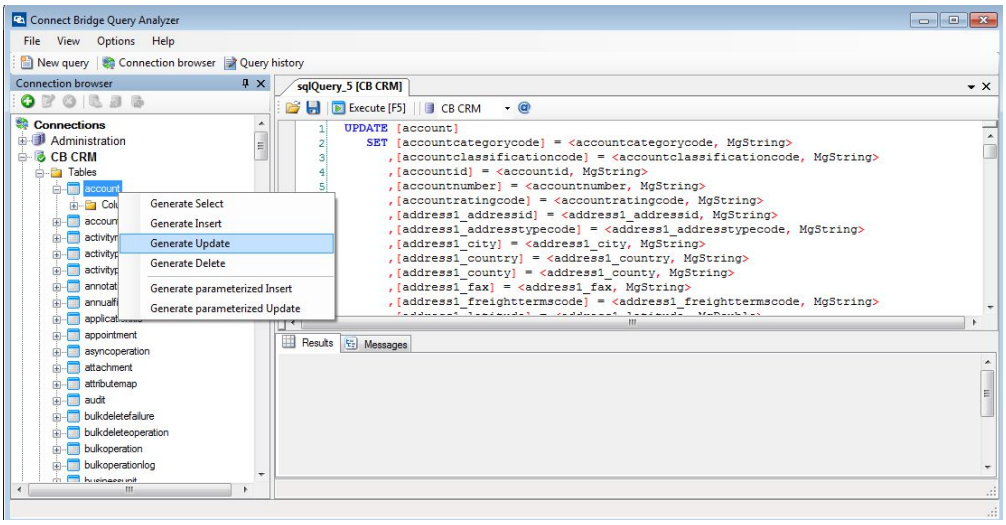


Figure 29 – Auto-generate SQL

This is particularly useful when INSERT-ing or UPDATE-ing more elaborate entities.

The difference between the parameterized versions and the ones without is that the parameterized ones will also auto-generate parameters to be used in the parameters window.

In case of **stored procedures**[12], auto-generate also works by generating either a parameterized or a non-parameterized version of the chosen stored procedure. Simply right-click on the appropriate one and choose (see Figure 30 - Auto-generate Stored Procedures). In case of the parameterized, simply add the appropriate parameters in the Values Column of the parameters window[13].

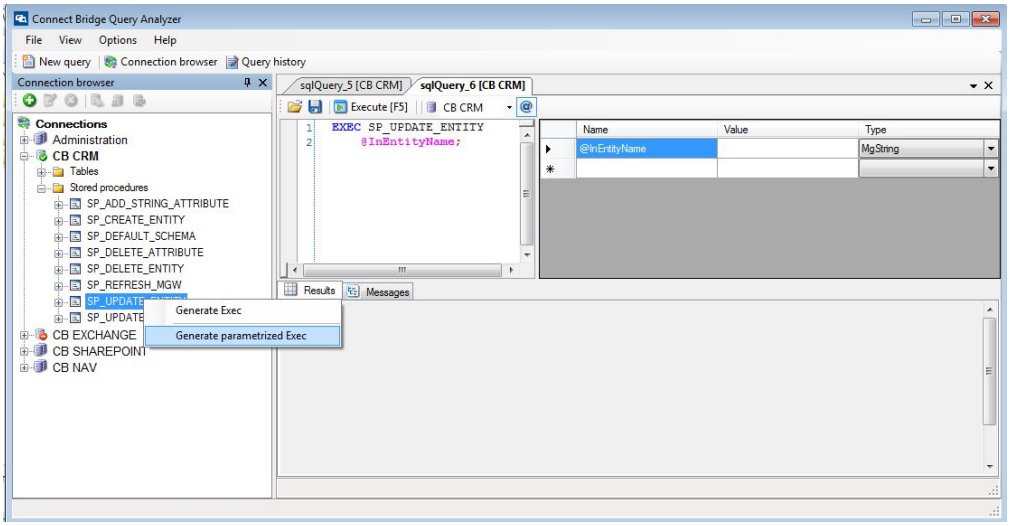


Figure 30 - Auto-generate Stored Procedures

In the sample shown in Figure 30 - Auto-generate Stored Procedures, the stored procedure SP_UPDATE_ENTITY (which update the schema of a table) has been right clicked, then Generate parameterized Exec was chosen and the stored procedure as shown in the Query Editor Window was auto-generated. After filling in Value as the value for the parameter @InEntityName in the Parameter Window on the right and pressing Execute [F5], the result will be shown below in the Results window.

4.2.3 View Results/Errors

Results and Errors (Messages) can be seen in their respective tabs of the lower section of the Query Editor Window (see Figure 19 - Query Analyzer: Standard View [4b] and Figure 25 - Query Analyzer: Expanded View [4d]. It might also be useful to check the Event Viewer Window (see Figure 25 - Query Analyzer: Expanded View [5]) by clicking on the Event viewer button (see Table 2 - Icon bar icons)

For a full reference of available stored procedures, please check the respective Connector reference or the registered users section of our developer's corner on www.connecting-software.com

The parameterized versions of the SQL statements are particularly useful when the statements are only tested in the QA and meant to be inserted into custom code later.

5 Workflow Development Tutorial

The following sample gives a quick tutorial on how to access the CB Server from your own code. Samples are given in C# and JAVA, but any programming language that can use ODBC, JDBC or Web Services can be used.

5.1 Scenario

In the given scenario, we will demonstrate a simple workflow by showing how to insert an appointment from MS Exchange (the source server) into MS SharePoint (the destination server), based on a condition (Flag "Private" = false). In this example, we are using ODBC for C# and JDBC for JAVA. For Web Services, please refer to the registered users section of developer corner on our webpage www.connecting-software.com

In order to implement our scenario (and handle any other workflow), the developer needs to follow these steps:

1. Create one ODBC/JDBC/Web Services connection to each target server
2. Provide an appropriate object like a DataTable in C# or a ResultSet in JAVA
3. Fill the object with data from the source server
4. Manipulate the data and or apply a workflow rule
5. INSERT or UPDATE the Data in the target server
6. Close connections

For simplicity's sake, we will hard-code connection data in the samples given below. **In real life, Connecting Software, however, recommends to use app.config or properties files and properly encrypt any sensitive information.**

5.1.1 C# sample

```
using System;
using System.Collections.Generic; using System.Data;
using System.Linq; using System.Text; using System.Data.Odbc;

namespace AppointmentWorkflowTutorialBasic
{
class Program
{
static void Main(string[] args)
```

```
{
// STEP 1. Create one ODBC/JDBC/WebServices connection to each target server
OdbcConnection exchangeConn = CreateMGWConnection("123.456.789.000", 8087,
"accountExchange",
                                                                    "demouser",
"'password'");

OdbcConnection sharepointConn = CreateMGWConnection("123.456.789.000", 8087,
"accountSharepoint",
                                                                    "demouser",
"'password'");

//prepare a select command and a data adapter

OdbcCommand selectExAppointmentsCmd = new OdbcCommand("SELECT * FROM
[Appointment];", exchangeConn);
OdbcDataAdapter selectExAppointmentsAdapter = new
OdbcDataAdapter(selectExAppointmentsCmd);

//STEP 2. Provide an appropriate object like a Data Table in C# or a ResultSet in JAVA
DataTable exAppointmentsDataTable = new DataTable(); Console.WriteLine("Connecting to
Exchange..."); exchangeConn.Open();

//STEP 3. Fill the object with data from the source server Console.WriteLine("Fetching
Exchange appointments..."); selectExAppointmentsAdapter.Fill(exAppointmentsDataTable);
Console.WriteLine("Found {0} appointments...",
exAppointmentsDataTable == null ? 0 : exAppointmentsDataTable.Rows.Count);

//open connection to exchange and process exchange appointments inserting them into
sharepoint
Console.WriteLine("Connecting to Sharepoint..."); sharepointConn.Open();

//STEP 4. Manipulate the data and or apply a workflow rule
//in this case, check if the appointment is private, if not
-> insert it into sharepoint
foreach(DataRow appointment in exAppointmentsDataTable.Rows)
{

if (!appointment.IsNull("IsPrivate") && (bool)appointment["IsPrivate"])
{
```

```
Console.WriteLine("Skipping \"{0}\"", appointment["Subject"]);
continue;
}

//create a new odbc command for inserting OdbcCommand insertToSharepointCmd =
new OdbcCommand("INSERT INTO [Calendar] ([Title], [Description], [Location], [StartTime],
[EndTime]) VALUES ( ?, ?, ?, ?, ?)",
sharepointConn);

//fill its parameters with values for the sharepoint account
insertToSharepointCmd.Parameters.AddWithValue("@Title",
appointment["Subject"]);
insertToSharepointCmd.Parameters.AddWithValue("@Description", appointment["Body"]);
insertToSharepointCmd.Parameters.AddWithValue("@Location", appointment["Location"]);
insertToSharepointCmd.Parameters.AddWithValue("@StartTime", appointment["StartDate"]);
insertToSharepointCmd.Parameters.AddWithValue("@EndTime", appointment["EndDate"]);
Console.WriteLine("Inserting \"{0}\"", appointment["Subject"]);

//STEP 5. INSERT the Data into the target server int affectedRowCount =
insertToSharepointCmd.ExecuteNonQuery();


}
Console.WriteLine("Closing connections...");

//STEP 6. Close Connections exchangeConn.Close(); sharepointConn.Close();
Console.WriteLine("Press any key to exit..."); Console.ReadKey();
}

/// <summary>
/// Creates an odbc connection to CB server
/// </summary>
/// <param name="host">CB server host name</param>
/// <param name="port">port the CB server service is listening to</param>
/// <param name="accountName">CB Account name (has configured credentials to log into
the remote systems)</param>
/// <param name="userName">CB User name</param>
/// <param name="password">CB password</param>
/// <returns></returns>
```

```
static OdbcConnection CreateMGWConnection(string host, int port, string accountName, string
userName, string password)
{
OdbcConnection conn; OdbcConnectionStringBuilder connStrBuilder = new
OdbcConnectionStringBuilder();
connStrBuilder.Driver = "Media Gateway ODBC Driver"; connStrBuilder.Add("ACC",
accountName); connStrBuilder.Add("UID", userName); connStrBuilder.Add("PWD",
password); connStrBuilder.Add("PORT", port.ToString()); connStrBuilder.Add("HOST", host);
connStrBuilder.Add("IMPL", "CORBA");

conn = new OdbcConnection(connStrBuilder.ToString()); conn.ConnectionTimeout = 240;
return conn;
}

}
}
```

5.1.2 Java sample:

```
import  java.sql.Connection;  import java.sql.DriverManager; import
java.sql.PreparedStatement; import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Program {

        public static void main(String[] args) throws SQLException,
ClassNotFoundException {

Class.forName("com.cnsconnect.mgw.jdbc.MgDriver");

                // STEP 1: Create one ODBC/JDBC/WebServices connection to each target
server
                String exchangeConnectionString =
"jdbc:MgDriver:IMPL=CORBA;ENC=UTF-
8;HOST=123.456.789.000;PORT=8087;UID=demouser;PWD='password';ACC=accountE
xchange;";
                String sharepointConnectionString =
"jdbc:MgDriver:IMPL=CORBA;ENC=UTF-
8;HOST=123.456.789.000;PORT=8087;UID=demouser;PWD='password';ACC=accountS
```

```
harePoint;";

            Connection exchangeConn =
DriverManager.getConnection(exchangeConnectionString);
            Connection sharepointConn =
DriverManager.getConnection(sharepointConnectionString);

                        Statement exchangeSt = exchangeConn.createStatement();
                        System.out.println("Connecting to Exchange...");
            //STEP 2: Provide an appropriate object like a Data Table in C# or a
ResultSet in JAVA
                        //STEP 3: Fill the object with data from the source server
                        ResultSet exchangeRs = exchangeSt.executeQuery("SELECT *
FROM [Appointment]");

                        //create a new JDBC statement for inserting PreparedStatement
                        sharepointSt =
sharepointConn.prepareStatement("INSERT INTO [Calendar] ([Title], [Description],
[Location], [StartTime], [EndTime]) VALUES ( ?, ?, ?, ?,
?)");

                    //STEP 4: Manipulate the data and or apply a workflow rule
                        //in this case, check if the appointment is private, if not
-> insert it into sharepoint
                        while (exchangeRs.next()) {

                    Boolean isPrivate = exchangeRs.getBoolean("IsPrivate");

                        if (isPrivate != null && isPrivate)
                        {
            System.out.println("Skipping '" + exchangeRs.getString("Subject") + "'");
                            continue;
                        }

// Title
    //fill its parameters with values for the sharepoint account sharepointSt.setString(1,
    exchangeRs.getString("Subject"));
```

Description

```
// Location
 sharepointSt.setString(2, exchangeRs.getString("Body")); // sharepointSt.setString(3,
 exchangeRs.getString("Location"));
 sharepointSt.setTimestamp(4,
            exchangeRs.getTimestamp("StartDate")); // StartTime
                        sharepointSt.setTimestamp(5,
            exchangeRs.getTimestamp("EndDate")); // EndTime


               System.out.println("Inserting '" + exchangeRs.getString("Subject") + "'");


                        //STEP 5: INSERT the data into the target server
                        sharepointSt.execute();
                         }


                         exchangeRs.close();
                         exchangeSt.close();
                         sharepointSt.close();


                         //STEP 6: Close Connections exchangeConn.close();
                         sharepointConn.close();
                    }
}
```

6 CB SQL Reference

This section provides a reference to the Connect Bridge version of the SQL Syntax. It closely resembles the ANSI-92 standard. Please check the registered users section of the developer's corner on our website www.connecting-software.com for the most up-to-date information.

6.1 General SQL Syntax Schema

{}defines a mandatory part

[]defines an optional part

|defines one option to be selected

[expression …] three dots define a potential repetition of the expression as many times as needed

All keywords and references are not case sensitive. You can write them in upper case or lower case.

Use the Connection browser in the CB Query Analyzer to identify the correct names of Tables and Columns for your statements.

Note: All SQL Examples are written for the MS Exchange Connector unless otherwise designated.

6.2 Supported Data Manipulation Statements:

6.2.1 SELECT

Reads the data from the connected server. It represents the "R" in CRUD[14]

SELECT {**column_reference** | **function_reference**} **FROM table_reference**

[**join reference**] [**WHERE condition reference**]
[**ORDER BY column_reference** {**ASC** | **DESC**}]
[**LIMIT** number [**OFFSET** number]];

column_reference - {**column_reference** [**AS** alias]} [**,** {**column_referenc** [**AS** alias]} ...]

Refers to the name of a column or its alias (i.e. shortcut). Use commas to separate columns. A column must be part of the table chosen in **table_reference**. If the column name conflicts with an SQL keyword, use brackets: e.g. **[**from**]**

An Alias is a string that can be sued instead of the column name and as an abbreviation

function_reference – see chapter 6.2.7 FUNCTIONS

table_reference refers to the name of the table chosen

CRUD is a general term denoting the most common commands in data manipulation: **C**reate, **R**ead, **U**pdate, **D**elete

join_reference – see chapter 6.2.5 JOINS

condition_reference – see chapter 6.2.6.CONDITIONS

ORDER BY defines the sort order of the returned list according to the given priority rule for ordering (first column has the highest priority).

ASC ascending sort order (A->Z)

DESC descending sort order (Z ->A)

Example:

```
-- Select all Contacts, show only the columns Surname & Given Name
(First name) and sort the list by surname from A to Z.
SELECT Surname, Givenname FROM Contact ORDER BY Surname ASC;
--Same example but with alias – please note that in case of aliases, the resulting
column headers will show the alias name SELECT Surname AS s,
Givenname AS g FROM Contact ORDER BY s ASC;
```

LIMIT designates the maximum number of rows to be returned

OFFSET designates the number of rows after the first that should be skipped before the selection begins

Example:

```
-- Returns the Surname, GivenName, Email & Phone Nunmber of
the conacts number 3,4 & 5 (OFFSET 2 means start with number
3 and LIMIT 3 means take three incuding the one you started with.
SELECT Surname, GivenName, Email1, TelephoneNumber FROM Contact LIMIT 3
OFFSET2;
```

value_reference

Any literal value used (in quotes) has to fit to the data type of the column the value belongs to. For more information take a look on supported data types. See the connection browser in the CB Query Analyzer for a respective list and chapter 6.2.8. SUPPORTED DATA TYPES for details

For example: the column BirthDate in Contacts requires a MgDateTime data type, trying to assign a simple number won't be enough.

Simple Example:

```
-- get the subject and body from all tasks
SELECT Subject, Body FROMTask;
```

Full Example:

```
-- Skip 20 messages and select 10 messages from the Inbox folder where the Subject is equal to „Test" and order the rows by the name of the sender from A to Z
SELECT DisplayName, [From], [To], Subject, [Body], DateReceived
FROM Message
JOIN Inbox ON Message.ID=Inbox.FKItemID
WHERE Subject='Test' ORDER BY [From] ASC LIMIT 10 OFFSET 20
```

6.2.2 INSERT

Inserts data into a target server. It represents the "C" in CRUD.

```
INSERT INTO table_reference ({column_reference}[, {column_reference} ...])
values ({value_reference}[, {value_reference} ...]); [Select Scope_identity();]
```

Scope_Identity returns the ID generated by the INSERT statement but only when used in conjuction with the INSERT statement.

Example:

```
-- Create a new contact and get the generated ID of this inserted record. INSERT INTO Contact (GivenName, Surname, City, Country, Department,Email1) VALUES ('John', 'Smith', 'NewYork', 'USA', 'unknown', 'John.Smith@domain.com'); SELECT Scope_identity();
```

6.2.3 UPDATE

Updates existing data on a target server with new information. It represents the "U" in CRUD.

```
UPDATE table_reference
SET column_reference = {value_reference}[,column_reference = {value_reference} ...]WHERE condition_reference;
```

Example:

```
-- Change the start time of appointment with name Appointment1 to start 30 minutes before current start time.
UPDATE Appointment SET StartDate='2012-01-07 12:30:00.000', EndDate='2012-01-07 16:30:00.000'
WHERE Organizer='user' AND DisplayName='Appointment1' AND StartDate='2012-01-07 13:00:00.000';
```

6.2.4 DELETE

Deletes existing data on a target server. It represents the "D" in CRUD.

WARNING! Unless certain precautions have been set in the accounts setup by the CB Administrator, "DELETE FROM Entity" without a WHERE clause will DELETE ALL DATA in that entity (e.g. "DELETE FROM Contact" will delete all contacts). It is therefore highly recommended to ALWAYS USE A WHERE CLAUSE WHEN DELETING!

```
DELETE FROM
table_reference WHERE
condition_reference;
```

Example:

```
-- Remove tasks with the name Task1 created on the shown date and time.
DELETE FROM Task
WHERE DisplayName='Task1' AND CreationDate='2012-07-05 08:30:00.000';
```

6.2.5 STORED PROCEDURES

Stored procedures are preconfigured sets of commonly used or useful functionalities, provided by Connecting Software. Please check out the respective Connector Reference or our registered users section of the developer's corner on our webpage www.connecting-software.com for a detailed description of all available stored procedures.

```
EXEC stored_procedure_name
[parameter_reference[, parameter_reference ...]];
```

stored_procedure_name

The name of the stored procedure supported by the database. You can get a list of all supported stored procedures by execution of following SQL statement:

```
EXEC SPSYSODBCPROCEDURES;
```

To get a list of parameters for some stored procedures, use following statement:

```
EXEC SPSYSODBCPROCEDURECOLUMNS stored_procedure_name;
```

parameter_reference

{**parameter_reference**}[**,**{**parameter_reference**} ...]

Refers to the comma separated list of stored procedure parameters referenced by value. Use single quotes to denote a parameter value. Make sure that the data type matches the expected data type (e.g. string, int, etc.).

Example:

```
-- Forward a sent mail message with an existing ID to a user with the mail address userTo@domain.com and prefix the subject of the mail with the string ‚FWD: ‘.
EXEC SP_FORWARD_MESSAGE '<CHANGE THIS TO EXISTING Message.ID>',
'body --- ', 'FWD: ', 'userTo@domain.com';
```

6.2.6 JOINS

Joins are meant to combine information from several tables based on conditions. They are needed whenever lookup tables are used or when trying to filter information based on conditions on other tables (e.g. finding all contacts in CRM that are a member of a certain account).

join_reference

{[**INNER** | **LEFT** | **RIGHT**] **JOIN**} **table_reference ON condition_reference**

The **JOIN** keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables. The Join operations are supported only if there is an explicit constraint defined between the tables. The left table stands before **JOIN**, the right table behind it.

JOIN	The same as **INNER JOIN**. The keyword **INNER** is optional.
LEFTJOIN	Returns all rows from the left table, even if there are no matches in the right table.
RIGHTJOIN	Returns all rows from the right table, even if there are no matches in the left table.
INNER JOIN	Returns rows if there is at least one match in both tables. If there are rows in the left table that do not have matches in the right table, those rows will NOT be listed and vice versa.

Examples:

```
-- Select all tasks with their task priority.
SELECT Subject, [Body], StartDate, DueDate, FkTaskPriority AS Priority
FROM Task LEFT JOIN TaskPriority ON Task.FKTaskPriority=TaskPriority.ID;

-- Select all appointments even if no importance is assigned to them. SELECT
DisplayName, [Body], FKAppointmentImportance FROM AppointmentImportance RIGHT
JOIN Appointment ON Appointment.ID=AppointmentImportance.ID;

-- CRM sample: Selects last name & first name of all contacts that are attached to an
account
SELECT lastname, firstname FROM contact AS c JOIN account AS a ON
c.parentcustomerid = a.accountid;
```

6.2.7 CONDITIONS

Refers to conditional values used in a WHERE clause.

condition_reference

{**column_reference operator_reference value_reference**} [**operator_reference** {**column_reference operator_reference value_reference**}

...]

operator_reference refers to one of the following standard operators:

< (smaller than), **>** (bigger than), **<=** (smaller or equal to), **>=** (bigger or equal to), **=** (equal to),

and, or, like

Comparison operators: <, >, <=, >=, =

are used to execute a comparison based on the data type of the column designated in the column_reference operand.

Example

```
-- Select all tasks that have a due date of Aug 31st or later
SELECT*FROMtaskWHEREDueDate >= '2012-08-31 00:00:00.000';
```

The like operator

can be applied on string operands only. It evaluates (to true or false) if the value of the column_reference contains a substring provided as the second operand in the condition.

Example

```
-- Selects all contacts that have a surname that contains the part"Do"
-- and shows the Columns Surname and Givenname. This would include a"John Doe"
-- but also a "James Done" or a "Charlene Randolph"
SELECT Surname, Givenname FROM Contact WHERE Surname LIKE'Do';
```

Logical operators: and, or

are used to construct more complex conditions; **and**required both conditions to be true in order to qualify, **or**requires one or the other

Example

```
-- Selects all contacts that have a surname that contains the part"Do"
-- and shows the Columns Surname and Givenname and the First Nameequals "John"
-- This would include "John Doe" but not "James Done" or"Charlene Randolph"
SELECTSurname, GivenName FROMContact WHERESurname LIKE'Do' and
GivenName = 'John';
```

Example

```
--Selects the subject of all appointments that start on or later thanAug 1st 2012 or
--end on Aug 31st or sooner
SELECTSubject FROMAppointment WHERESTartDate >= '2012-08-01
00:00:01.000' orEndDate <= '2012-08-31 00:00:01.000';
```

6.2.8 FUNCTIONS

Standard SQL functions that allow manipulation of data.

functions_reference

{function_name**(parameters_reference)**}[**,**
{function_name**(parameters_reference)**} ...]

Function_namerefers an item of the following list of supported functions:

Upper	Converts all characters in a string to upper case.
Lower	Converts all characters in a string to lower case.
Substring (column_reference**,** start-index, endindex**)**	Returns a substring of the original string starting with character number startindex and ending with character number endindex.
Len	Returns the length of the string.
Row_number()	Retrieves a sequential numbering of rows
Count(*)	Counts all records in a table which fulfill a criteria
Scope_identity()	See insert statement.

Function parameters are handled in an equivalent manner to stored procedures parameters (see Stored Procedures).

Examples:

```
-- Returns a capitalized list of surnames
SELECTUPPER (Surname) FROMContact;

-- Returns the characters of the surname starting with character 2 and endign with character 4 e.g. from „Smith" it would return „mit"
SELECTSubstring (Surname, 2, 4) FROMContact;

-- Returns the length of each surname
SELECTLen (Surname) FROMContact;

-- A common use is the combiantion of these 2: select all thecharacters starting with the third to the last (in case of „Smith" it wouldreturn
„ith"
SELECTSubstring (Surname, 3, Len(Surname)) ASName FROMContact;

-- Returns a sequential number of the row for use in developementlogic
SELECTRow_number() ASNumber, Surname FROMContact;
-- Count all folders
SELECT COUNT(*) FROM Folder;
```

6.2.9 DATA TYPES

The following table provides a list of data types supported by the CB:

Data Type	Description
Numeric Types	(signed) byte, Int16, Int32, Int64, Single, Double, Decimal
String	'any string'
Boolean	trueor false
Char	Single Character e.g. 'a'
DateTime hh:mm:ss.000'	Data and Time in the format: 'yyyy-mm-dd e.g.'2012-08-01 13:15:00.000
Byte array	Array of bytes.

Table 5 - Supported Data Types

7 Glossary and Abbreviations

7.1 Glossary

Term	Explanation
Client-Server	It is distributed application which partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.
Communication port	Any software that uses the Internet, exchanges data between your computer and other computers connected to the Internet. Depending on the type of information exchanged between the software of your PC and the other computers of the Internet, specific communication channels also called "communication ports" are used. Communication ports are associated to the numbers, e.g. for SMTP is the port 25.
Connection String	It is a string that specifies information about a data source and the means of connecting to it. It is passed in code to an underlying driver or provider in order to initiate the connection.
Daemon	In multitasking computer operating systems, a daemon is a computer program that runs as a background process, rather than being under the direct control of an interactive user. In Windows OS daemons are called services.
Data Source	In our meaning it is a database, which holds data.
localhost	It is the standard hostname given to the address of the loopback network interface.
CB Administration Tool	It provides functionalities to manage and configure Connecting Software Connect Bridge.
CB Server Service Controller	It controls CB-Server Service. It allows user to start and stop CB-Server Service. The tool is installed on the server machine.
CB-Driver	One of following three drivers: CB-ODBC Driver, CB- JDBC Driver, CB-Web Service Driver

CB-JDBC Driver	JAR file created by Connecting Software Connect Bridge developers in the java programming language. It communicates with the CB-Server via TCP/IP protocols. CORBA mechanism is used for communication between CB-JDBC Driver and CB- Server Service processes.
CB-ODBC Driver	DLL library created by Connecting Software Connect Bridge developers in C/C++. It communicates with the CB-Server via the TCP/IP protocols. CORBA is used as a communication mechanism.
CB -Connector	The connector, which provides the access to one or more of external data sources.
CB -Server	Server component of CB
CB -Server Service	It is a daemon running in the Windows system as a service, where Connecting Software Connect Bridge is installed. It listens for SQL statements from client application, processes the statement and provides a result (usually requested data).
CB -Web Service Driver	CB-Web Service Driver is installed on the server machine and it runs under the Internet Information Server (IIS). It provides API interface accessible via Proxy Web Service to the Client Application.
Query Analyzer	It is a client tool designed to access data provided by Connecting Software Connect Bridge.
SQL statement	It is a SQL language element, which may have a persistent effect on schemata and data, or which may control transactions, program flow, connections, sessions, or diagnostics. It also include the semicolon (";") statement terminator. Though not required on every platform, it is defined as a standard part of the SQL grammar.
Stored Procedure	It is a subroutine available to applications that access a relational database system. Stored procedures can consolidate and centralize logic that was originally implemented in applications.

7.2 Abbreviations

Abbreviation	Meaning
API	Application Programming Interface
CB	Connect Bridge
CORBA	Common Object Request Broker Architecture
CRM	(Microsoft Dynamics) Customer Relationship Management
DLL	Dynamic Link Library
EULA	End-User License Agreement
EX	Microsoft Exchange
FTP	File Transfer Protocol
GUI	Graphical User Interface
HDD	Hard Disk Drive
HTTP/S	Hypertext Transfer Protocol / Secure
HW	Hardware
ID	Identification
IIS	Internet Information Services
IP	Internet Protocol
JDBC	Java Database Connectivity
JRE	Java Runtime Environment
MS	Microsoft
NAV	Microsoft Dynamics CRM
ODBC	Open Database Connectivity
OS	Operating System
PC	Personal Computer
QA	Query Analyzer
RAM	Random Access Memory
SP	Microsoft SharePoint
SQL	Structured Query Language
SW	Software
TCP	Transmission Control Protocol
XML	Extensible Markup Language