1.1 Welcome to ZOC Terminal

What is ZOC Terminal

ZOC is a professional terminal emulator which lets you access character based hosts via SSH, Telnet, Modem and other means of communication. It can be used to connect to Unix/Linux hosts and shell accounts, IBM mainframes, routers, microcontrollers, BBS mailbox systems, internet muds and many more.

In addition to widely used terminal emulations such as Linux, Xterm, VT220, TN3270 and several types of Ansi, ZOC also supports terminal emulations such as Linux and Sun's CDE and well-known file transfer protocols such as X, Y, Zmodem, SCP and Kermit.

ZOC is highly configurable and offers some advanced and unique features. It supports features such as tabbed sessions, a powerful scripting language, automatic language, automatic triggering of actions based on received or typed text or typed text or data tracing features, to name a few.

Quick Start Guides

- Secure Shell (SSH) Connections
- Telnet Connections
- Direct Serial Connections
- Modem Connections
- Scripting/Programming/REXX
- System/Network Administrators (Installation/Deployment)

Other Themes to Start With

- ZOC Command Line Parameters
- Screen Elements
- Menu Functions
- Orders/Updates/Contact
- Common Questions (How-To Guides)
- Common Problems
- Features you might have missed
- Uninstalling ZOC

1.1.1 Quick Start Guides

ZOC is a professional terminal emulator which lets you access character based hosts via Telnet, Modem, SSH, ISDN and other means of communication.

Below you find a list of tutorials about how to get started using ZOC with various communication mechanisms and about other common tasks.

- ❑ Secure Shell (SSH)
- ❑ Telnet
- ❑ Direct Serial Connections
- ❑ Modem
- ❑ Scripting/Programming
- ❑ System/Network Administrators (Deployment)

1.1.1.1 SSH Quick Start

Connecting to Servers

If you want to connect to a host via Secure Shell (SSH), select File Menu→Quick Connection.

This will bring up a dialog where you can select Secure Shell from the list of connection types.

Then enter your destination and select the desired emulation (Xterm or VT220 will work in most cases). The destination in the Connect to box can either be a host name or IP address, for example `ssh.hogwarts.edu` or `192.168.1.1`

If you are connecting to a server which is running on a port other than 22 (the default for SSH), you can provide an alternate port number by entering it in the Port field. Leaving the port field empty, will use the SSH default port.

If you want to change terminal characteristics like window layout, colors, etc. click the Edit button to change options in the session profile. (Session profiles are predefined sets of options which can be used as the underlying configuration for a connection.)

Creating Desktop Links

In the Quick Connect dialog, note the Save As button which allows you to save a connection as a desktop icon, as an icon in the button bar or as an Host Directory entry.

Host Directory

To put a SSH host into the host directory go to File Menu→Host Directory and click New.

Enter a title for the entry and select the Secure Shell connection type. Then enter the host name or IP Address in the Connect to field. You should also specify an emulation (e.g. Xterm, Linux or VT220).

If you want ZOC to automatically log you in, you can specify the user name and password in the Login tab.

Security Warning

Saving your password will allow anyone who has access to ZOC to connect to the host without entering a password. Use this feature only in conjunction with a ZOC program password Options→Program Settings→Passwords.

File Transfer via SSH

To transfer files over SSH connections, please use SCP or Zmodem (see the description for the Secure Shell/SSH).

Other Related Topics

Read more about other SSH related functions like options, host names, ports etc.

Also read about the /SSH command line parameters which lets you start ZOC with an SSH session from the command line. Last but not least please click the Help button in the dialog where you edit entries in the host directory.

1.1.1.2 Telnet Quick Start

Connecting to Servers

If you want to connect to a host via Telnet, select File Menu→Quick Connection.

This will bring up a dialog where you can select Telnet from the list of connection types.

Then enter your destination and select the desired emulation (VT220 or Xterm will work in most cases). The destination in the Connect to field can either be a host name or IP address, for example `myhost.hogwarts.edu` or `192.168.1.1`

If the server you are connecting to is running on a port other than 23 (the default for telnet), you can provide an alternate port number by entering it in the Port field. The port value can be a number or a well known service name, e.g. `smtp` or `pop3`. Leaving the port field empty, will use the telnet default port.

If you want to change terminal characteristics like window layout, colors etc, click the Edit button to change options in the session profile. (Session profiles are predefined sets of options which can be used as the underlying configuration for a connection.)

Creating Desktop Links

In the Quick Connect dialog, note the Save As button which allows you to save a connection as a desktop icon, as an icon in the button bar or as an Host Directory entry.

Host Directory

To put a Telnet host directly into the host directory go to File Menu→Host Directory, click New.

Enter a title for the entry and select Telnet as your connection type. Then enter the host name or IP Address in the Connect to field. You should also specify an emulation (e.g. vt220 or Linux).

To have ZOC remember your login procedure for later automatic login, enable Record keystrokes on next login in the Login tab.

Other Related Topics

Read more about other Telnet related functions like options, host names, ports etc. Check the /TELNET command line parameters. Find out how to configure ZOC as the web browser's Telnet client in how-to questions. Last but not least please click the Help button in the dialog where you enter the host directory entry details.

1.1.1.3 Direct Serial Connections Quick Start

Using Quick Connect to Access Devices Attached to a Serial Port

If you want to talk to devices which are directly connected to a serial port (e.g. cisco routers, microcontrollers, etc.), select File Menu→Quick Connection. In the quick connect dialog then select

Serial/Direct as the connection type.

In the Connect to field enter the text `localhost` or the text `serial`. Then click the Configure button for Serial/Direct, check Override session profile and choose a com port via the Scan button and set the serial parameters.

Alternately you can type the actual name of a com port in the Connect to field. In this case, the value will override the port which is specified by the Configure button.

Important: Do not leave the Connect to field empty. You need to provide the text `localhost` or the text `serial` or the name of a serial port there.

Connecting to Serial Devices via Host Directory

You can also use the host directory for these type of connections. Go to File Menu→Host Directory, click New.

Enter a name for the entry and select Serial/Direct from the list of connection types. The Connect to field and Configure button are used in the same way as with the Quick Connect dialog (see above).

Important: Do not leave the Connect to field empty. You need to provide the text `localhost` or the text `serial` or the name of a serial port there.

To have ZOC learn and remember your login procedure for later automatic login (assuming the attached device requires a login), enable Record keystrokes on next login in the Login tab.

Serial/Direct vs. Serial/Modem

The Serial/Direct connection type is similar to Serial/Modem in many respects. However, the main difference is that a 'connection' with Serial/Direct refers to the connection between the computer and a *local piece of equipment* (e.g. a router), which is directly attached to a local serial port or serial usb-port. Serial/Modem however defines 'connection' as the connection between the computer and a *remote location*, where the attached equipment (usually a modem) is merely a way to establish the desired connection. .

In other words, use Serial/Direct if you want to talk to the locally attached device (router, microcontroller, etc.) itself. Use Serial/Modem if the attached device is a modem which understands AT commands and which you want to use to connect to a remote computer.

More about the Serial/Direct and Serial/Modem connection types can be found in Serial/Direct

1.1.1.4 Modem Quick Start

Connecting to Remote Servers

If you want to make a Modem connection, select File Menu→Quick Connection.

This will bring up a dialog where you can select either Windows Modem (recommended) or Serial/Modem (if you are running under macOS or if you want to deal with the modem's AT commands through the com port directly).

You can modify the options (select the Modem or COM port to use) by clicking the Configure button next to the connection type. Then enter the phone number and select the desired emulation.

The destination in the Connect to box will be the phone number to connect to (it will be dialed through the modem via AT commands).

If you want to change terminal characteristics like window layout, colors etc. click the Edit button to change options in the session profile. (Session profiles are predefined sets of options which can be used as the underlying configuration for a connection.)

Creating Desktop Links

In the Quick Connect dialog, note the Save As button which allows you to save a connection as a desktop icon, as an icon in the button bar or as a Host Directory entry.

Talking to the Modem Directly

If you want to send AT commands to a modem directly, or if you have hardware (e.g. a router) which directly attached to a serial port, you can bypass the Quick Connection dialog. Edit the Session Profile (Options Menu) instead, and select Serial/Modem as your connection type. You will then be able to type commands to that port immediately after you close the session profile dialog.

Also see the Direct Serial Connections Quick Start for a different way to make connections to devices which are connected directly to a serial port.

Host Directory

To put a new modem host directly into the host directory go to File Menu→Host Directory and click New.

Then enter a name for the entry and select Serial/Modem, or Windows Modem (recommended) as the connection type. You can modify the options (select the Modem or COM port to use) by clicking the Configure button next to the list of connection types. Then provide a phone number in the Connect to field.

To have ZOC remember your login procedure for later automatic login, enable Record keystrokes on next login in the Login tab.

Serial/Direct vs. Serial/Modem

The Serial/Direct connection type is similar to Serial/Modem in many respects. However, the main difference is that a 'connection' with Serial/Direct refers to the connection between the computer and a *local piece of equipment* (e.g. a router), which is directly attached to a serial port or serial usb-port. Serial/Modem defines 'connection' to be a connection between the computer and a *remote location*, where the attached equipment (usually a modem) is merely a way to establish the desired connection.

In other words, use Serial/Direct if you want to talk to the locally attached device (router, microcontroller, etc.) itself. Use Serial/Modem if the attached device is a modem which understands AT commands and which you want to use to connect to a remote computer.

Other Related Topics

More about the Serial/Modem and Serial/Direct can be found in Serial/Modem and Windows Modems. Also, if you are using Serial/Modem, see modem configuration.

1.1.1.5 Scripting/Programming Quick Start

If you want to use scripting to automate tasks with ZOC you can learn more about ZOC's scripting language in the following topics.

- ❑ Introduction to REXX Programming (highly recommended)
- ❑ REXX Language Elements
- ❑ ZOC-REXX Commands.

If you want to run scripts from the command line, please check the `/RUN:` and `/U` parameters in command line parameters.

Do not miss our samples in your ZOC documents folder (File Menu→Show Data Folder) plus the REXX resources on http://www.emtec.com/zoc/documents.html#rexxfiles

Dynamic Data Exchange (Windows) or AppleScript (macOS) may also be of interest.

1.1.1.6 Deployment, System/Network Administrators Quick Start

If you are a system or network administrator and if you want to set up ZOC for users in your organization the following topics will be of interest in order to configure and organize the deployment.

If you have questions beyond these topics or specific requirements, please do not hesitate to ask our support.

Command Line Options

ZOC offers a set of command line parameters to run it in special modes or make it perform operations directly after startup.

When checking the command line options, the options `/CONNECT`, `/CALL`, `/RUN`, `/O` may be the ones of immediate interest.

Command line parameters can be put into the shortcut icon which starts ZOC or into a file named commandline.ini (the latter could be part of a custom installation, see below).

Configuration Files

If you need to replicate ZOC configurations throughout your organization, the necessary files will be HostDirectory.zhd (host directory), Standard.zfg (program options), *.ZOC (session profile), *.ZKY (key maps) and maybe the *.ZRX files. All these files are located in the ZOC Data Folder, which can be accessed via File Menu→Show Data Folder.

Additionally the file Admin.ini should be of special interest if you want to tailor a ZOC installation in a shared environment.

Also note the newuserprofile_english and newuserprofile_german folders in the ZOC program folder. These contain the default configuration which will be used (copied) to create the initial configuration for a user on his first start.

Licensing

Usually software license codes will be stored in the Windows registry under HKLM/HKCU+Software+EmTec+ZOC9+Registration. Alternately ZOC offers file based licensing and license counting which can be used with custom or network installations (see below). Please contact EmTec support for details.

Network Installation/Multi User Installation

ZOC can be installed on a network drives and can use shared or individual configurations in read-only or write-access mode. ZOC can be tailored to use host directory files, options files or options directories in various locations by using the Admin.ini file (see the ZOC directory for a sample) and/or by defining the directories in Options→Program Settings (see also Data Folder Selection).

Some tailoring is also possible by using command line options together with a `commandline.ini` file although we strongly recommend to use the `Admin.ini` file whenever possible.

If necessary, EmTec Support will help you to find the best network installation configuration and licensing code handling option for your individual environment.

Building a Custom Setup

You can modify the ZOC installation to fit your special needs. The main installation file (zoc_nnn.exe) is a self extracting ZIP archive. You can unzip into a temporary directory (e.g. Z:\CDINST) with a standard unzip tool like INFOZIP.

There you will find two files named SETUP.CFG and SETUP.FIL. SETUP.CFG is the installation script and controls the program folder name and the icons in the Windows start menu. SETUP.FIL is a ZIP archive, which contains a complete ZOC program directory including default configuration files. You can replace the default configuration files in the newuserprofile_english folder or add your own files into this archive (see User Data Folder).

Alternately you can create an additional ZIP archive named CUSTOM.FIL which will be unpacked into the program directory after the files from SETUP.FIL have been extracted.

Then you can burn the Z:\CDINST directory on CD or make it available to licensed users in your organization as a LAN resource. In both cases, the custom version will be installed when the users run setup.exe.

Unattended Installation

The ZOC setup does not support unattended installation. However, you can simply replicate a ZOC program directory from a previous installation to other PCs and create the necessary icons/shortcuts with your own deployment tools.

Questions?

If you have any further questions regarding installation, custom versions or licenses please contact Markus Schmidt at EmTec for Support.

1.1.1.7 Uninstalling ZOC

Windows

The Windows version of ZOC comes with an uninstall program. You can uninstall either from the ZOC program group in the start menu or via the Add/Remove Software in the Windows Control panel.

If you want to manually remove the ZOC files, remove the ZOC program folder C:→Program Files→ZOC9, the ZOC data folder My Documents→ZOC9 Files and the registry entries Software→EmTec→ZOC9 in HKEY_CURRENT_USER and HKEY_LOCAL_MACHINE.

macOS

On macOS the ZOC program and its configuration files are stored in the following locations:
/Applications/zoc9.app
/Library/Preferences/com.emtec.*
~/Library/Application Support/ZOC*Files/
~/Library/Logs/EmTec*
~/Library/Preferences/com.emtec.*
Searching for ZOC in Finder or Spotlight should find all these files.

1.2 Starting ZOC (Command Line Parameters and Environment)

Topics in this section:

1.2.1 Command Line Parameters

There is a wealth of parameters that can be specified in the command line, in the icon's parameter field (Windows only) or via one of the Command Line Files (Windows and macOS). These parameters are mostly intended to make ZOC perform tasks automatically after starting.

Note: In addition to calling ZOC with the command line parameters below, you can also use the command line conversion tools `ssh.exe` and `telnet.exe` which are located in the ZOC program folder and which accept command line parameters compatible with OpenSSH and Windows Telnet. For more information, please start these tools with a `-?` parameter.

```
ZOC [/CALL:<host name>] [/CONNECT=<connection-type>!<host>]
[/DEV:<connection-type>] [/EMU:<emulation>]
[/FI:<fontname>] [/FX:<fontname>]
[/FORCEASIANFONTS] [/FORCENOASIANFONTS]
[/KEY:<file>]
[/LANG:<language>]
[/MIN] [/MAX] [/NUMLOCK:<n>]
[/MOUSEPOS:<x>,<y>]
[/O:<seesion profile>]
[/HOSTDIR:<host directory file>]
[/PROGCFG:<program settings file>]
[/RESTRICT:n]
[/RUN:<scriptfile>] [/RUNARG:<text>]
[/RLOGIN:<host>[:<port>]]
[/SSH:[<login>@]<host>[:<port>]]
[/SSHUSER:<ssh-username>]
[/SSHPASSWORD:<ssh-login-password>]
[/SSHKEY:<private keyfile>]
[/STANDALONE]
[/TABBED]
[/TELNET:<host>[:<port>]]
[/TT:<file>] [/U] [/WD:<workdir>]
[/WINPOS:<x>,<y>,<width>,<height>]
```

Parameters can be entered from a Windows command prompt or added to the target field of a Windows desktop shortcut icon. For example, if you wanted to create an icon that starts a ZOC connection, right click the desktop, choose New→Shortcut and enter something like:
`"c:\program files\zoc9\zoc.exe" /TELNET:hogwarts.edu`

Please note the quote characters which are necessary because of the space character in "Program Files". Parameters themselves need also to be enclosed by quotes if they contain space characters, e.g.
`"c:\program files\zoc9\zoc.exe" "/CALL:New Mail Server"` or
`zoc "/EMU:Ansi BBS"`

Normally the current working directory is ignored by ZOC (see Working Directory), but in command line parameters you can use the directory which was the working directory at startup, by using the placeholder *%CURDIR%*, e.g. `/O:%CURDIR%\test.zoc`

As an alternative to the Windows command line parameter format (as indicated above), it is also possible to specify command line parameters in Unix style, parameters starting with dash rather than the Windows typical slash and providing parameter values separated by a space character rather than using the colon. Thus the following command line calls are equivalent:

`ZOC /MIN /RUN:test.zrx` or `ZOC /MIN "/CALL:Office Router"`
`ZOC -min -run test.zrx` or `ZOC /MIN -call "Office Router"`

Description of Commandline Parameters

/CALL:hostentry
This is a name of a host from the host directory to be called after startup, e.g. `ZOC /CALL:CoolServer`

/CONNECT=<connection-type>!<host>
This option allows the caller to connect to an arbitrary host via an arbitrary connection method (device) from the command line.

The connection type parameter specifies the name of a connection method as listed in ZOC's connection dialogs. The host parameter is a destination in the context of the connection method, e.g. an internet host name, ip-address or telephone number (host names can have a port number appended to them after a colon).

If the connection type is SSH (Secure Shell), you can specify an username and password in the form *<username>:<password>@<host>*,e.g.
`ZOC /CONNECT=SSH!joedoe:secret@myhost.secure.com`
(For authentication with key files from the command line, please see the `SSH` and `SSHKEY` parameters below.)

For the RLOGIN communication you can use *<username>@<host>*.

/DEV:connection-type
You can use this option to tell ZOC to activate a specific connection method at startup. Supply the name of the connection type as shown in Options→Session Profile→Connection Type, e.g.
`ZOC /DEV:TELNET`

If necessary, you can append connection type sub-options (as described in the ZocSetDevParm REXX command) to the connection type name, separated by an @ character:

`ZOC "/DEV:SERIAL/MODEM@COM3:57600-8N1|9|350"` (please be aware that strings which contain the vertical bar character must be enclosed in quotes).

/EMU:emulation
This command line parameter activates the given terminal emulation, e.g. `ZOC /EMU:VT220`

/FI

Normally ZOC checks the whole list of fonts to see which fonts it can use. This list might include fonts that you do not want to use. When started with the /FI (Font-Inclusion) option, ZOC will only use fonts, who's names match the name given with /FI, for example if you start ZOC as `ZOC "/FI:Courier New"`, ZOC will only offer the Courier New font in its font list. You can select multiple fonts this way.

Important: You should make sure that at least one of these fonts is available and suitable for use with ZOC, otherwise ZOC will exit with a Panic message.

/FX

The /FX option is similar to /FI, except that ZOC will use all suitable fonts, except the ones listed with /FX, for example to use all fonts except the Terminal fonts, start ZOC as `ZOC /FX:Terminal`.

/FORCE[NO]ASIANFONTS

Under Windows ZOC excludes specific Asian fonts (like SimSun or BatangChe) from the font list if the system language is English or German. With these options, the Asian fonts are forced to be included or excluded.

/KEY:file

This option tells ZOC to load a specific key mapping file at startup, for example `ZOC /KEY:Alternate.zky`

/LANG:language

This option can be used to switch the program language between German (`/LANG:GER`) and English (`/LANG:ENGL`).

/MIN

Start ZOC with the window minimized.

/MAX

Start ZOC with the window maximized.

/MOUSEPOS:x,y

Moves the mouse to the given pixel position, e.g. `/MOUSEPOS:1024,768` (Windows only)

/NUMLOCK:n

You can use this option if you want to start ZOC with the NumLock key set to on (n=1) or off (n=0). (Windows only)

/O:file

With this option, you can tell ZOC to load a specific session profile instead of the normal Standard.zoc file: `ZOC /O:BlackGlass.zoc`

/HOSTDIR:file

This is the name of an alternate host directory to load. The startup default is HostDirectory.zhd

/PROGCFG:file

Using this option allows you to specify an alternate name for the file that contains the program settings (default is Standard.zfg).

/RESTRICT:n

This option allows you to restrict access to certain parts of ZOC (for example if you use ZOC in an pre customized environment).

This option is provided for backward compatibility only, please use the Admin.ini file instead (see the ZOC program directory for an example).

/RUN:scriptname

With this option, you provide a REXX program to be executed after start, e.g. `ZOC /RUN:PROCESS.ZRX`. ZOC will look for the file in the designated scripts folder (Options→Program

Settings→Folders), ZOC configuration folder (`ZOC9 Files`) or program installation folder.

Alternately a path can be specified, for example `ZOC /RUN:C:\SCRIPTS\NIGHTCALL.ZRX`. Please note that in case of paths that contain space characters, you will need to enclose the whole expression in quotes: `ZOC "/RUN:C:\My Scripts\AutoCall.zrx"`.

If the REXX program expects a parameter which accessed inside the script as `parm= ARG(1)`, the argument can be provided via `/RUNARG` like this: `ZOC /RUN:DOIT.ZRX "/RUNARG:HELLO WORLD"`

Note: This /RUN parameter can not be used together with other parameters that initiate a connection (e.g. /TELNET, /CONNECT, ...). In these cases please use the /RUN parameter and initiate the connection from within the script via `ZocConnect` or similar commands.

Note: Please also check the /U parameter below.

/RLOGIN:dest

Short form for */CONNECT=RLOGIN!<username>@<host>:<port>* (see above), e.g. `ZOC /RLOGIN:harry@hogwarts.edu`

/STANDALONE

Under the Windows operating system, the parameter will cause the ZOC window to separate itself from other ZOC windows. Technically this means that the window and its sessions will run in their own process space (e.g. in the task manager you will see one zoc.exe per window). This means that a crash in one window will not affect others, but this also limits some functions, e.g. tabs can not be dragged/dropped between windows.

Note: In order to affect all instances of ZOC, this parameter should typically be provided through the commandline.ini file.

/TABBED

When this parameter is provided on the command line together with a parameter that initiates a connection (e.g. `/CONNECT`, `/TELNET`, `/SSH`, etc.) or together with the `/RUN` parameter, it will attempt to open a new tab in an existing ZOC window. Otherwise (without /TABBED) a new window for each call. Also, you can use the /TITLE:<name> parameter in conjunction with /TABBED to set a name for the new tab.

/SSH:dest

This is a short form for */CONNECT=SSH![<username>[:<password>]]@<host>:<port>* (see above).

Syntax variants are

```
ZOC /SSH:sshsrv.hogwarts.edu or
ZOC /SSH:harry@secure.hogwarts.edu
ZOC /SSH:harry:secret@ssh.hogwarts.edu:10022
zoc -ssh harry@secure.hogwarts.edu
zoc -ssh harry@secure.hogwarts.edu 10022
```

If the username or password contains special characters, that conflict with this syntax, use the `/SSHUSER`, `/SSHPASSWORD` and `/SSHKEY` parameters below.

/SSHUSER:username

Instead of including the username with the `/SSH` parameter (see above), you can provide the username as a separate parameter. This is especially helpful if the username contains special characters that might conflict with the *user:password@host* syntax.

/SSHPASSWORD:password

Instead of including the username with the `/SSH` parameter (see above), you can provide the password as a separate parameter. This is especially helpful if the password contains special

characters that would conflict with the */SSH:user:password@host* syntax.

/SSHKEY:file

In conjunction with the `/SSH` parameter, it is possible to provide the name of a private key file by using the `/SSHKEY` parameter. The value can be either the name of a private key file in the ZOC SSH directory or a fully qualified path:
`ZOC /SSH:harry@secure.hogwarts.edu /SSHKEY:id_dsa`

Alternately the key file can be added to the `/SSH` parameter directly, by appending it to the password with an extra colon:
`ZOC /SSH:harry:secret:id_dsa@secure.hogwarts.edu` or
`ZOC /SSH:harry::id_dsa@secure.hogwarts.edu`

/TELNET:dest

This option is a short form of */CONNECT=TELNET!@<host>:<port>* nd tells ZOC to connect to a Telnet host after starting up, e.g. `ZOC /TELNET:docsrv.hogwarts.edu`
`ZOC /TELNET:192.168.0.5:110`
`ZOC -telnet login.hogwarts.edu smtp`.

/TT:file

This option tells ZOC load another character translation table at startup, for example `ZOC /TT:SCAND.ZTR`.

/U

This option specifies unattended mode. In unattended mode ZOC suppresses request boxes like "Are you sure" and uses their default answers instead.

This way the program will not hang if something unforeseen happens when you run it in the middle of the night to fetch your mail.

/WD:folder

This option specifies the working directory where data files are expected (options, etc.). See also Data Folder Selection.

/WINPOS:x,y[,width,height]

The /WINPOS parameter allows to control where and how large the ZOC window will open. x/y are screen pixels from the upper left corner, optional width/height are the width and height of the window.

Upon startup, ZOC can modify the width/height values according to the layout options in the session profile (e.g. to match the required matching font size).

1.2.2 ZOC Command Line Files

commandline.ini

If you want some of the command line parameters to be always used, you can create a file named `commandline.ini` using an editor and put *one command line option per line* into the file.

Under Windows save the file in the ZOC program directory (same folder where ZOC.EXE resides).

Under macOS create a file named `.zoc9_commandline` in your home folder (i.e. `/Users/You/.zoc9_commandline`).

Example:

```
commandline.ini:
/RUN:mystart.zrx
/MAX
```

ZSH/ZLN Files

If you want to work with multiple different sets of command line options (e.g. to make connections from the command line) you can create files with the extension `.zsh` or `.zln` and fill them with *one command line option per line*.

This file extension is automatically associated with ZOC; a double click on such a file will launch ZOC with the file and will thus make it start using the options which are contained within it.

Note: Under macOS using the .zsh extension may be in conflict with other applications, so using .zln is preferable there.

Example:

```
Server7.zln:
/DEV:RLOGIN
/EMU:VT220
/CONNECT:harry@rsh.mydomain.de
/WINPOS:20,20,750,500
```

1.2.3 ADMIN.INI/Working Directory/Network

Initial Working Directory

When starting ZOC from a Windows command prompt or from an icon, the current directory (from within which you issue the command) will be ignored. Instead, all primary program files are loaded from the directory where ZOC was installed.

Data files are loaded from the location which is configured in the Admin.ini file (see also Information for System/Network Administrators and User Data Folder Selection) or from the `/WD:` command line parameter (if that parameter is provided).

If you want to reference the current working directory when specifying files with command line parameters, you can use the placeholder `%CURDIR%`, e.g. `/RUN:%CURDIR%\script.zrx`

The ADMIN.INI File

The Admin.ini is stored in the ZOC9 Program folder (Windows) or in the Contents folder inside the zoc9.app Bundle (macOS). Under macOS the location to place a custom admin.ini file is ~/.zoc9_admin.

On both operating systems you will find a template for this file in your TEMP folder.

This file determines default values for the program settings file (mainly default directories) and can limit the program functions which the user can access (e.g. preventing access to the program options).

The Admin.ini file is the primary start point that determines the default user directories. From the locations specified there, ZOC will load secondary configuration files (program settings, session profiles, ...) and user files (scripts, uploads, ...).

With this file it is also possible to create per user configurations, mixed configurations (shared and per user folders) or fully shared and preconfigured configurations.

Please read the sample Admin.ini file in the ZOC program folder or in the TEMP folder for documentation about the settings. See User Data Folder Selection for a description of the exact process how ZOC determines the user data folder.

1.2.4 Data Folder Selection (Workdir)

ZOC stores its configuration files in a central folder, which is usually located in the User's "My Documents" folder.

If the data folder does not exist (because ZOC is started for the first time or because the folder was moved), the program will ask if a new folder should be created or where an existing folder is located. In case of new folder creation, ZOC will copy the files from the HIBLUR(newuserprofile_english) folder in the program directory to the user's new data folder.

The basis for this selection is the `ConfigDataFolder` setting in the Admin.ini file which is also located in the ZOC program directory. For single user installations, this setting will always point to the user folder (unless a working folder was specified on the command line via `/WD:` parameter.)

If the user changes the folder (e.g. by renaming it and then pointing ZOC to the new folder), the new selection will be written back to the Admin.ini file.

For shared installations (installations, where the Admin.ini file contains a setting of `SharedInstall=yes`), the Admin.ini will be used to specify the default location of the user data folder. However, if the user changes the location of his data folder, the new location will not be written to the Admin.ini file (because that would affect other users). Instead, the user's choice will be stored in the user registry (Windows) or user preferences com.emtec.zoc9.plist (macOS) under `HKCU+Software+EmTec+UserConfigFolder`

The possibility of relocating the folder in a shared installation through a registry entry can be disabled in the Admin.ini file by setting `DisableRegistryConfigFolder=yes`. (The option to specify the working folder on the command line can also be disabled in the Admin.ini file.)

This means that, unless the option is disabled, ZOC will first look for the `/WD:` parameter. If that is not present (or disabled), in a shared installation in which the registry config specification has not been disabled, ZOC will then check if a `UserConfigFolder` entry is in the registry. If that entry is there it will use that value as the folder specification. If the installation is not shared or if the `DisableRegistryConfigFolder` option is true, the location of the user data folder will always be taken from the Admin.ini file.

In the next step, ZOC will load the program settings file, which by default is named `Standard.zfg` and resides in the `Options` subfolder of the user data folder, but which can be in an other location as per specification in the Admin.ini file. This file will then point to the secondary data folder, where script files, logs, etc. are stored.

Although being a bit complex, this process allows maximum flexibility in defining the data structure and ranges from fully shared installations (where the user data folder points to a shared folder on the network), to mixed installations (where some folders are shared and other are per user) or fully individual data storage (user data folder in the user's "My Documents" folder, with a `Standard.zfg` which has sub folder settings which are pointing also to locations in the user's realm.

1.2.5 Environment Variables

ZOC uses the TZ environment variable to correct the file date transmitted by Zmodem.

Zmodem was developed with Unix in mind, where time stamps (file date/time) are always stored in Greenwich time and where local time is calculated by the dir command. This way a file always carries the correct local time, even if shipped on disk around the globe. However, the HPFS file system knows nothing about the time zones which are still part of Zmodem. Most MS-DOS systems emulate the time zone in one way or another, mostly by using Eastern Standard Time (5 hours west of Greenwich) as a default; files received from such senders get the correct time when received by ZOC. However, some hosts use other time zones. If you want to adapt your time zone setting to such hosts, you should add the

TZ environment variable to your CONFIG.SYS.

To do this use SET TZ=<SSS><N><DDD> where <SSS> is a 3-char standard time code, <N> is the time difference from Greenwich and <DDD> is a 3-char daylight saving time code, for example use SET TZ=EST5EDT for Eastern Standard Time (which is the default) or TZ=CET2CDT for Central European Time.

1.2.6 Startup REXX Programs

There are two REXX programs that are started when you run ZOC. The first one is `ZOCEVENT.ZRX` (it is also executed when ZOC closes) and it is intended to stop or restart a fax program's (for example "FaxWorks") auto receive feature. See the file `ZOCEVENT.TXT` for more details.

The other REXX program is called `STARTUP.ZRX` and is run after ZOC has been initialized. You can use it to call a host every time you start ZOC or print startup greetings or things like that. Please note that the file is not executed if you have command line options for another script or command line options which would immediately result in a connection (e.g. `/CALL` or `/CONNECT` or similar).

1.3 Screen Elements

The elements of the screen are:

- Menu
- Toolbar
- Custom Button Bar
- Main Window
- Local Typing
- Status line

1.3.1 Menu

The menu lets you select most of the ZOC functions (for a description of the particular functions see Menu Overview).

1.3.2 Toolbar

The toolbar offers icons for most of the functions in ZOC.

You will see a description for each icon, if you move the mouse over it and wait two seconds.

You can Customize the Toolbar by clicking with the right mouse button on empty space in the toolbar.

1.3.3 Custom Button Bar

The custom button bar is located beneath the toolbar and offers buttons that can be defined to send text, make connections, start REXX programs etc.

The buttons are part of ZOC's session profiles (Options→Session Profile→User Buttons). You can assign different session profiles to different connections to have different sets of buttons depending on which host you connect to.

In the button options you can assign labels for the buttons and if you move the mouse to a button and wait a second, the full text (or action code) of the button will be shown.

1.3.4 Main Window

The main window displays the received text. The characteristics of the text (size, font, colors) are configured in the Session Profile (Options menu).

By default you can bring up a popup menu by clicking the right mouse over the main window.

You can mark text with the left mouse button (this is called stream mode).

If you hold the Alt key pressed while marking text, the selection will be made in block mode rather than in stream mode (block mode lets you select arbitrary rectangular areas on the screen, stream mode is line

orientated).

If you hold the Ctrl key pressed while doing the selection, the text will be appended to the text that already is in the clipboard rather than overwriting it.

The marked text is copied into the clipboard and can be accessed from the functions Paste, Paste (no line breaks), Paste Quoted, Paste CIS-Quoted, Print Clipboard and Edit Clipboard in the Edit Menu and with some key shortcuts.

1.3.5 Local Typing

The Local Typing line is an entry field into which you can enter text to be sent to the host. The text is editable and it is not sent until you press Enter. This is helpful if you are in an online conference or if you use a remote shell that does not provide command editing and history.

The Local Typing line can be activated from Options→Session Profile→Window Parts or by pressing Alt+C or by pressing the Scroll-Lock key (the latter, only if defined accordingly in Options→Session Profile→Terminal Options).

If the cursor is in the Local Typing line, you can discard it by pressing ESC or Scroll-Lock key. With the line open Alt+C will toggle the cursor between the main window and the Local Typing line.

You can use arrow keys to retrieve the last few lines you typed (this works like the command history in an Windows shell).

1.3.6 Status Line

The status line provides information about and access to some options (if, because of a small font, your ZOC window is not wide enough, some of the information will not be available).

Placing the mouse on one of the status line elements (fields or LEDs) for about two seconds will show a short description.

Connection Type and Parameters

This button displays the currently selected connection type and for serial connections it will display line speed and parameters. Clicking on the field with the left mouse button brings up the Options→Session Profile→Connection Type dialog. There you will be able to select and configure the I/O methods (connection types).

Emulation

This button displays the current terminal emulation. Clicking on the button brings up the Options→Session Profile→Emulation dialog which will let you choose from the list of available emulations.

Transfer Protocol

This button displays the currently set file transfer protocol. Clicking on the field brings up the Options→Session Profile→File Transfer dialog where you can select and configure transfer protocols.

LEDs

The four LEDs in the status line provide two services:
If they turn green, your host has requested the VT102 LED service and sets the LEDs according to the application you run. The LEDs will stay green until you change the emulation or select the Clear Screen/Reset function from the View Menu.
The LEDS can be turned on/off in Options→Session Profile→Window Parts.
If they are red they display status information:

LED 1

Bright Red: Script running.

LED 2

Blue: Doorway mode active, Pink: AutoLogin running, Green: Learning REXX or AutoLogin.

LED 3/4

Modem receive-data and send-data.

Log File

This checkmark displays the name of the log file and opens or closes it.

Online Time

This shows how long you are (or have been) connected to a host.

Window Size

This shows the size of the window (columns x rows) or the current cursor position (configured in Options→Program Settings→Window).

1.4 Menu Commands

The menu provides some submenus which give you access to ZOC's functions:

- File Menu
- Edit Menu
- View Menu
- Logging Menu
- Transfer Menu
- Script Menu
- Tools Menu
- Options Menu

1.4.1 File Menu

The file menu manages all necessary functions to maintain the communication methods (Modem, Telnet or ISDN, etc.) and to connect to a host. Additionally, selected host directory entries can be dialed directly from this menu.

Additionally some methods dependent function may appear in this menu (like Telnet's `Are you there`). They are described in the method's help text as well).

Quick Connection
This function opens a dialog to allow you to connect to a host as required, without creating a Host Directory entry (see Quick Connect).

Host Directory
The host directory is a dialog to maintain your personal dialing directory or to connect to one or more numbers from it (see Host Directory).

Reconnect
If a connection was terminated, this function will connect to the same host again.

Disconnect
Disconnects the current connection.

New Window
Opens a new ZOC window.

New Tab
Opens a new tab within the ZOC window.

Close Window
Closes the current ZOC window.

Print Screen
This function lets you print the contents of the terminal window (in raw ASCII mode).

Save Screen

Saves the contents of the terminal window to the file defined in Options→Program Settings→Special Files. If the file already exists, the text will be appended.

Accept Connections

This function enables/disables the auto answer mode of your chosen communication method. The Telnet device will create an incoming port with port number 23. The Serial/Modem handler will send the commands you set up in Session Profile→Connection Type→Serial/Modem→AT-Commands→Modem Options.

In auto answer mode the modem will automatically pick up the phone when the phone rings and will try to establish a data connection with the calling modem (of course this is not a good idea when you expect voice calls).

Stop AutoLogin

This function aborts automatic login initiated from a host directory entry (see Changing Host Directory Entries). AutoLogins are indicated by the 2nd LED in the status line being pink.

Call Next in Queue

If you select more than one host to call from the host directory, ZOC will connect to the first of them that is available. After finishing the call you can select this function to try to reach the next available host.

Reset Communication

This function will reset the connection type to its default state. For modems this means that the init string defined in the Modem Options will be sent to the modem to set it to a defined state. It will do this regardless of the current carrier detect state.

Send Break

Pulls the data pin of the modem low for a given time (see Options→Session Profile→Connection Type→Serial/Modem). This is called sending a break. Some systems use this to reset the connection state or to stop the current operation. Devices other than the modem will perform a similar operation or will just ignore this command.

Connection Type Dependent Functions

Some ZOC communication methods (e.g. Telnet) add method dependent functions. These are described in the input/output devices section of the help file.

Connect XXXX

At the end of the file menu appear those entries from the host directory that have the Menu Access option enabled (see Changing Host Directory Entries). This is handy for hosts which you call often.

1.4.2 Edit Menu

The edit menu allows you to access your computers' clipboard. Normally it is copied from a ZOC screen by marking characters with the left mouse button (this can be combined with the Alt key), but it can be copied or cut from an editor or any other program as well.

See also: Options, Program Settings, Program Settings→Clipboard

Copy

This function copies the selected text into the clipboard.

Copy Window

This function copies the entire contents of the ZOC text window into the clipboard.

Copy Scrollback Buffer
Copies the content of the whole scrollback buffer into the clipboard.

Paste
This function takes text from the clipboard and sends it to the host. The sending is done like a text upload and the delay and CR/LF translation options from Options→Session Profile→Text Sending are used.

Paste (no line breaks)
Takes text from the clipboard and sends it to the host. Carriage Returns are ignored and a Space is sent instead. This function is affected by the options in the Options→Session Profile→Text Sending dialog.

Imagine the following case: You searched a host for files with the keyword TERM and the host gave you a list of filenames - one in each line. To download all files, you could mark the list with a box (left mouse button while pressing the Alt key) and send them with a space between them when the host asks for the filenames to download.

Paste continue
If a paste operation is stopped because the text did not fit on the screen, it can be continued from the point where the stop occurred. (This function is only available when the TN3270 emulation is used.)

Clipboard to Editor
Calls the system editor (see Options→Program Settings→Special Files) with the contents of the clipboard.

Print Clipboard
Sends the contents of the clipboard to the printer (the printer is configured in Options→Program Settings→Printer).

1.4.3 View Menu

In the View menu you will find functions that have to do with screen display etc.

Probably the most used function will be the scroll back view. It is a memory buffer that stores the lines that scrolled off the top edge of the screen. A similar function is the Data Stream Viewer, which captures and views data exactly in the order it was received or to show Session Profile→Trace/Debug data.

See also: Logging to File functions

Scroll Back
This function shows the contents of the scrollback buffer inside the main window to see text that has scrolled off the screen.

Scroll Back as Window
Alternately to scrolling back inside the main window, it is possible to open a scrollback in a separate window, thus being able to see current and scrolled text at the same time. The font for the window is selected in Options→Program Settings→Scrollback

Find in Scroll Buffer
Searches and highlights text within the scrollback buffer.

Clear Scroll Buffer
Clears the capture buffer. This function is affected by the Options→Program Settings→Confirmation.

Scroll Buffer to Editor

Saves the scrollback buffer to file and opens the file in an editor.

Show Data Stream Viewer
Shows the data stream viewer window. The data stream viewer is similar to scroll back, but shows session data in the order that it was received. It can also show trace/debug data (e.g. to show incoming data as hex, see Session Profile→Trace/Debug).

Clear Data Stream Buffer
Clears the content of the data stream viewer (see above).

Next/Previous Tab
Switches to the previous/next tab within the window.

Tabs as Thumbnails
Shows the session from all tabs in miniaturized form.

Window Elements
This shows a submenu in which you can check the window elements for ZOC's main window. This is the same as in Options→Session Profile→Window Parts.

Snippets
ZOC monitors your input data stream for filenames and Internet addresses and collects them in a little window that floats aside the ZOC window. The visibility is also an option in Options→Session Profile→Window Parts.

Using this function you can show or hide the Snippets Window. When it is showing, you can send one of the collected items by double clicking it with the mouse.

Clear Screen/Reset Terminal
This function just clears the terminal screen and resets the LEDs in the Status Line back to red.

Editor
Call the editor which is defined in Options→Program Settings→Special Files.

Command Shell
Call the shell which is defined in Options→Program Settings→Special Files.

Local Typing
This function shows an entry field on screen where you can type locally before sending the whole line to the remote host (local typing uses the ASCII send options in the Session Profile, transfer-2). You can jump between the main screen and the Local Typing window via Alt+C and you can close the Local Typing window via ESC. If you have more than one tab, Alt+A toggles the "Send to all tabs" checkmark.

Split Chat
The split chat function splits the screen horizontally and shows text you type in the lower half while displaying text you receive in the upper. This is handy if you want to chat with someone who called you by modem directly.

Note: You should not use this function while accessing a remote host.

1.4.4 Logging Menu

ZOC maintains multiple different logging methods. There are two memory logging mechanisms which are found in the View Menu. The log menu itself features permanent logging functions, i.e. logging to disk file and printer.

The so called Log File that logs all incoming data permanently to a file on the hard disk. The other is redirecting all incoming data directly to the printer. The last one, the call log, logs time, duration, cost and file transfers of your calls to a file (see Options→Program Settings→Special Files→Calls Log).

The Logging Menu controls the capture, log file and printer logging.

Log to File

This function changes the toggles the status of logging between on and off. This can also be achieved by clicking on the capture name in the Status Line and can be done remotely if your host supports the DC2/DC4 logging protocol (see Options→Session Profile→Logging options).

Log to Printer

Sometimes it is handy to have part of a session sent to the printer. This is possible by enabling this function before receiving the data (e.g. before reading a mail).

Additionally it is possible to print already received text, by opening the capture window (see Logging Menu), marking the text with the mouse (to copy it to the clipboard). and printing the contents of the clipboard from the Edit Menu.

Set Logfile Name

Set the name of the current log file. The name can be automatically selected from the host directory when connecting to a host (when starting ZOC the log file gets its name from Options→Session Profile→Logging options). You may use special placeholders for the name as described in placeholder codes and in Options→Session Profile→Logging.

Use Default Logfile Name

This sets the log file to the file defined in Options→Session Profile→Logging options.

Delete Log Files

Since you can build log file names that contain the current date you might end up with a lot of log files. Using this function lets you delete log files you do not need any more. Selecting this function will open a file requester in which you can select one or more files which will be deleted if you press OK.

1.4.5 Transfer Menu

The transfer menu allows you to send files to the host or receive them from it. The functions in this menu use settings from Options→Session Profile→Transfer and Options→Program Settings→Folders in the Options Menu.

Related topics:

- SCP Transfer
- Zmodem Transfer
- IND$FILE Transfer
- Secure Shell/SSH
- SSH File Transfer
- File Transfer Window

Upload

Sending a file to a host is called an Upload. To do this the sender and receiver use certain methods called Transfer Protocols. The upload function sends a file to the host using the currently active protocol (see Options→Session Profile→File Transfer options).

Download

Receiving a binary file from a host is called a Download. The same transfer protocols are used as for uploads. Downloads are affected by the Options→Session Profile→File Transfer options.

Send Textfile

This function takes text from a file and sends it to the host without using a file transfer protocol. Roughly spoken, an ascii upload is the same as typing text very fast. This way you can prepare messages using an editor before you call a host and then send the file with this function when the host asks you to type your message.

Note: If the host loses characters in this process or if the transfer is too slow, you can modify a delay for each character in the Options→Session Profile→Text Sending dialog.

Send Binary File

The Send Binary function is very similar to the Send Text File function. The only difference is that the data in the file is not translated in any form and that the host is not expected to echo the characters back to ZOC. This function uses the character delay from Send Text, but ignores the line delay (see also: Options→Session Profile→Text Sending).

FTP Connection

If you are connected to a SSH or Telnet server or if you made a connection through the host directory, where you have defined an FTP connection associated with the connection, you can choose this function to open an FTP window. This window will allow you to navigate local and remote folders and to transfer files or group of files easily between your local computer and the remote host via drag and drop.

1.4.6 Script Menu

Start REXX Script

Starts a REXX script. REXX is something like a batch programs for ZOC, which lets you process specific tasks automatically (see ZOC Automation).

Stop Script

Stops a currently running REXX script.

Edit REXX Script

Call the editor to modify a REXX script manually.

Encrypt REXX Script

Encrypts a script file in a way that it can not be read by users while still being executable by ZOC. This is useful if you are distributing scripts to customers. (Please be aware this is just a lightweight encryption method that can be reversed by an averagely talented hacker.)

Record REXX Script

Most of the time REXX is used automate logins. Writing these is a rather tedious and boring task that can, for the most part, be done automatically. If you select this function before calling a host, ZOC will monitor the login process and create a REXX program that handles the login automatically.

Note: The program will only contain the code necessary for the log in. It will not contain commands to call that host (which is normally done from the host directory and not from script).

Stop Recording
After you enabled script learning and logged into a host you tell ZOC to stop the learning process. ZOC will ask you for a name and store the script accordingly. The name you use can later be entered into the REXX field of a host directory entry (see Changing Host Directory Entries). Then ZOC will execute it to log you in after having established a connection to that host.

1.4.7 Tools Menu

This menu contains tools which are not directly related to the current session but which are more of a housekeeping nature, e.g. tools to create and organize SSH encryption keys or to manage the stored passwords.

1.4.8 Options Overview

The options menu is used to define the characteristics of ZOC. There are basically two kinds of options: global program settings and variable settings (profiles).

Program Settings
Options which remain the same for all the hosts you call, e.g. directory settings, handling of connections, etc.

Session Profile
Options which may vary from host to host, e.g. connection type, terminal emulation, etc.

Keyboard/Character Translation Profiles
Profiles which are stored independent of the session profile and which can be loaded or attached to host directory entries.

Usually the various profiles are selected in the host directory entries Options tab or loaded from modified desktop icons via command line parameters. Profiles named `Standard` are automatically loaded at ZOC startup.

See also: Options Menu and Configuring ZOC, Session Profiles, Key Map/Translation Profiles, Program Settings.

1.4.9 Options Menu

See also: Options Overview

Program Settings
Configure the Program Settings which are likely to remain the same for all your connections.

Edit Session Profile
Configure session related settings (see Session Profile).

Jump to
Display a submenu to jump to a specific page session profile dialog.

Load Session Profile
Load a file of session profile.

Save Session Profile
Save the current session profile to disk.

Save Session Profile As
Save session profile to a file with different name. If saved as `STANDARD.ZOC`, it will be loaded automatically when ZOC is started.

Reset Session Profile
Reset the current session profile set to the default state.

Edit Keyboard Profiles
Brings up the Keyboard Profiles dialog which will let you load, modify and save keyboard profiles.

Edit Translation Profiles
Brings up the Character Translation dialog which will let you load, modify and save character translation tables.

1.4.10 Quick Connection

Note: We offer various Quick Start Guides, which provide detailed descriptions of this dialog, focusing on specific communication methods such as SSH (Secure Shell), Telnet and 10224,serial connections). In contrast, the text below offers a more general overview.

Quick Connection is a window that allows you to quickly set up a connection to a host. This feature is designed for connections that you do not plan to make regularly. For hosts you connect to frequently, you should add an entry to the Host Directory. Host Directory entries can be activated through various shortcut methods, such as user-defined buttons or key combinations, and they offer additional benefits like automatic login.

To make a quick connection, select a connection type, the host to connect to, an optional port number (for Telnet, Rlogin, and SSH connections), an emulation, and your login data (for SSH and Rlogin connections).

The connection will be based on various configuration options (e.g., window size, colors, cursor shape, etc.) defined in a Session Profile. The session profile also contains options for connection type and terminal emulation, but in the Quick Connection window, you can override these specific settings using the Configure buttons without altering the session profile (altering the session profile may affect other connections which are are based on that).

Connect to
Depending on the connection type, you can enter a host name or IP address, a phone number or whatever identifies the remote host.

This field offers also list of recently made connections. If you want to reconnect to a host to which you recently made a connection, simply pick the host from the drop down list.

Connection Type Specific Considerations for the Connect-To Field

Local Shell
For Local Shell you can use either localhost or 127.0.0.1. Alternately enter the name of a shell

to run in the ZOC window in shebang format, e.g. `#!/bin/bash` or `C:\Windows\cmd.exe`

Telnet, Rlogin or SSH

If the Telnet, Rlogin or SSH is selected for communication (see Connection type below), you will also be able to enter an ip port (e.g. 110 to connect to a pop3 server or a custom port number if your host requires it). Leaving the field empty will use the default ports for the respective method (e.g. 22 for SSH or 23 for telnet).

Serial/Direct

If you want to connect to a router or device which is directly attached to the serial port, there are two alternatives:

1. In the Connect to field or type `localhost` or `serial`. Then click the Configure button for Serial/Direct and select your serial port and communication parameters.
2. In the Connect to field enter the name of the communication port (e.g. `COM2` under Windows or `/dev/cu.pl2303serial` on macOS), then click the Configure button for Serial/Direct, set your parameters, but leave the port field in the configuration dialog empty.

Serial/Modem

If you connect through Serial/Modem, you will be offered to choose a Dial-Command. These commands refer to the AT-commands in the Serial/Modem options (click 'Configure' next to Serial/Modem, the edit the serial parameters and AT-Commands). The AT commands will be sent to the attached modem to make the actual connection.

Session profile

All connections in ZOC are based on a Session Profile. Session Profiles contain a combination of options (e.g., window size, font, colors, f-keys, etc.) that affect the session and all other sessions based on that profile. The Session Profile also includes settings for connection type and emulation. However, you can override one or both of these settings here in the Quick Connection window by choosing a different connection type and emulation. This allows you to use a Session Profile that generally meets your needs (in terms of font, screen layout, function keys, etc.) and reuse it for different connections, even if they require different connection types and emulations.

See also: Customizing ZOC

Connection type

Many of these methods can be customized by clicking the Configure button adjacent to the list, which will take priority over the same settings in the Session Profile. Alternatively, you can opt to utilize the settings configured for that connection type in the Connection-Type section of the Session Profile.

See this link for list of connection types and the options available for them: Session Profile→Connection-Types.

Emulation

Here you select a terminal emulation which will be used to let the host display data in the terminal area of ZOC. The emulation will depend on your remote host. For connections to Linux or Unix hosts the Xterm or VT220 emulations will be a good choices, although other systems sometimes require quite specific and even obscure choices. Emulation specific settings can be configured by clicking the Configure button next to the list and general defaults can be defined for each emulation in Options→Session Profile→Emulation.

Username

SSH and Rlogin communication will offer a field to enter your username. Type a username or use the text `%USERNAME%` or `%USERNAMELWR%` or `%USERNAMEUPR%` to log in using the same username that was used to log on to your workstation (the latter two placeholders are variants of the username translated to lower or upper case characters). This field will be used to identify you on the remote

host. If you enter ? as a username here, you will be prompted for one at login time.

Password

SSH will also offer a field to enter your password. The password is usually required together with your username and you will be prompted for one if you do not enter it here.

If you enable the Save password option, ZOC will remember the password if you use this dialog again for the same host (you can manage saved passwords in Options→Program Settings→Passwords).

SSH key

For SSH connections, certain hosts require files containing public and private encryption keys instead of relying on a username/password combination for authentication. You have the option to select an individual file from the SSH directory (configured in Program Settings→Folders) or use the global SSH authentication files or use keys from an ssh-agent. To configure that, click the Configure button next to where you select Secure Shell further up in this window, or access the Secure Shell settings in the Session Profile→Connection-Types section.

If, instead of a private key file, you provide the full name and path of a PKCS#11 compatible DLL or .dylib, ZOC will use that pkcs11 DLL to access your smart-card and retrieve the identy from the SmartCard chip.

Note: Public/private key file pairs can be created through the Manage SSH Keys function, which is available from ZOC's Tools menu.

1.5 Host Directory

The host directory can be accessed from the File menu and acts as a personal host directory for commonly used hosts and their settings.

The main window lets you select one or more hosts to call and functions to make connections or to maintain the list of hosts.

The Host Directory Options can be used to configure list display. You can also create multiple folders inside each of the sections (right mouse button).

Connect

Connects to one or more selected entries (you can select more than one entry by holding the Ctrl key pressed while you click on one entry after the other).

If multiple entries are selected, the action taken depends on the settings in the individual host directory entries. Entries can be configured to open a new tab or new window and entries are also configured to either use a dialog with retry-option to connect or to just make the connection without showing a dialog.

ZOC will try to launch parallel connections to as many hosts possible, but because the program can only show one AutoConnect dialog at a time, depending on the combination of entries and depending on their settings, ZOC will sometimes queue hosts to be called later via the Call Next function from the File menu. If you want to avoid queuing, set the hosts to open a new tab or window and/or use the SimpleConnect feature (in the Host tab of the individual entries).

New

Creates and edit a new entry in this section of the host directory.

Edit

Changes the data of the selected entry (see Changing a Host Directory Entry for details).

Clone

Creates and copy of the selected entry and opens the edit window.

Delete

Deletes one or more entries from host directory.

Create Shortcut

Lets you create a shortcut to a host directory entry on the Windows desktop or in the user button bar of the current ZOC session profile.

Move entry to

Moves selected entries between host directory sections.

Manage Sections

You can add, remove sections, change their order and define your own names for the sections of the host directory. (The sections are stored together with the entries in the host directory file ZOCHOSTS.INI.)

Import

Imports host directories from other comm programs.

Print

Prints the selected entries.

Mark Dues

Selects all entries which are due for calling (marked with a little yellow flash in front of the phone number). ZOC determines the entries that are due for calling by looking at the date of the last call and the 'Call after xx days' field from the edit window.

Settings

Show the host directory options which let you configure the columns, sections and other host directory related settings. See Host Directory Options.

1.5.1 Configuring a Host Directory Entry

A host directory entry combines the configuration profiles, which are managed from ZOC's Options Menu (see also Configuring ZOC) and which are stored in profile files.

Therefore the characteristics of a host directory based connection are a combination of Program Settings, Session Profile, Key Map, Translation Table, plus the selections for connection type, emulation, file transfer protocol, character set (code page), upload/download folder which you make in the host directory entry directly.

In this combination, the selections, which are made specifically in the host directory entry dialog (e.g. connection type, emulation, download folder) will override the corresponding settings from the file based profiles (e.g. host directory's connection type will override the session profile's connection type, host directory's download folder will override program settings' download folder, etc.).

Each entry consist of the following settings

- Host
- Options
- Login
- FTP
- Shortcuts
- Window
- BBS
- Info

1.5.1.1 Host

Title

A name for the entry. This name is shown in the host directory and it is used to set the title of the session's tab and it is shown in the main window's title when a connection is made.

Section

Select the host directory section to store this entry to. The sections can be managed in Host Directory Options.

Icon

You can choose a colored icon for an entry. The entry will display the icon in the host directory and it will also use the icon for the ZOC window when a connection was made. The color of the icon will also determine the color of a session's tab in the main window.

Sort first in host directory list

The host directory can be sorted by clicking on the column headers in the host directory main window. This option is intended for important hosts and will keep the entry always sorted near the top of the list.

Connect to

The internet address, hostname or phone number of the host. The specifics depends on the connection type (see Connection type below). The Quick Start Guides will show you how to set up connections for SSH, Telnet, serial cables, etc.

For Serial/Modem connections you can use one or more of the host directory macros in this field. These can be used to insert strings for calling cards, etc. into the phone number (see Options→Program Settings→AutoConnect options).

Dial Command

If you are using Serial/Modem as the connection type, you can select a dial command (from the Modem Options) to be used to dial the phone number.

Make Connection via ...

Here you choose if you want to use the AutoConnect feature (a connect window with retry option) or if you want to make the connection with a single attempt and no extra dialog (the latter behavior is similar to most standard command line tools like ssh or telnet).

The latter is useful for network based connections (SSH, Telnet, etc.) and it allows you to connect to multiple hosts at once by selecting multiple entries in the host directory main window and clicking Connect. If the entries have the option set to open a new tab when connecting (see host directory entry Window options), ZOC will open multiple tabs and concurrently connect to the hosts.

If however the hosts are configured to connect via AutoConnect, ZOC will instead show one connect dialog and will try to connect to the first available destination. Further hosts can then be called with the Call Next function from the File menu.

See also: Connect function in Host Directory Window

Session profile

Select the name of a session profile for use with the connection (e.g. Standard.zoc or TN3270.ZOC). Session profiles are preconfigured sets of settings (Fonts, Colors, Keys, Terminal behavior, etc) that can be shared among multiple connections which need similar options.

This way you can have entirely different options for different hosts or you can use the same options for a group of similar hosts. New options files can be created by editing one of the profiles and using the Save As button in the session profile dialog.

After the file was loaded, the connection method, emulation, file transfer protocol and code page will be set from the settings in this host directory entry, i.e. if necessary these options from the host directory entry can override the corresponding settings in the session profile.

See also Host Directory Entry Configuration

Connection type

This field lets you specify a communication method to make the connection.

For the selected connection type you will be able to configure that communication type, by clicking the Configure button next to it. For example with Secure Shell you can set different encryption methods for different hosts in your host directory. The settings dialog will vary with the selected connection methods.

The selection and options can override the selection of connection type and its settings in the

session profile (the session profile to use with this connection is specified further up in this dialog).

See also Host Directory Entry Configuration

Emulation
Select an emulation to use for the connection. The button next to the emulation list lets you change options which are specific to each emulation.

The choice of emulation and its options can override the emulation selection and its settings for that emulation in session profile (the session profile to use with this connection is specified further up in this dialog).

See also Host Directory Entry Configuration

1.5.1.2 Options

Keyboard profile
Select a keyboard profile (created via Options→Keyboard Profiles) to be used with the connection.

Translation profile
Select a character translation table (created from the Options→Translation Profiles) to be used when connecting to the host.

File transfer protocol
Choose a protocol to select when the connection is made (this choice can override the protocol selected in Options→Session Profile→File Transfer).

Font codepage
The code page represents the coding of characters in the selected font (with respect to the transmitted codes). It mostly required for the correct display of international characters and line graphics. This setting can override the code page selection in the session profile.

See Options→Session Profile→Layout for more information.

Logfile
This is a name to be used for logging the session. The name can contain absolute or relative folder paths (folders in the path, especially when containing placeholders, will be created as necessary).

Placeholders for date, time, etc. can be used as described in Options→Session Profile→Logging.

The initial logging state (active/inactive) is determined by the session profile.

If the field is left empty, the logfile name specified in the session profile will be used.

Download/Upload folder
This is the name into which downloaded files are stored (or from which uploaded files are picked). Unless the fields are left empty, they will for the duration of the connection override the values from Options→Program Settings→Folders.

1.5.1.3 Login

Username
Here you can set a username for login at your host.

The text `%USERNAME%` or `%USERNAMELWR%` or `%USERNAMEUPR%` can be used to log in using the same username that was used to log on to your workstation (the latter two placeholders are variants of the username translated to lower or upper case characters). On Windows computers, the placeholder `%DOMAINNAME%` is also available, which represents the domain or workgroup associated with the login.

Alternately, if the field contains a @-character immediately followed by a filename, e.g. `@C:\users\you\username.txt` or `@%ZOCFILES%\username.txt`, then the first line of text in that file will be used as a username.

This field will only be used if you are using Secure Shell or Rlogin as the connection method for this host, or if you are recording AutoLogin sequences (see below).

Password

You can add a password for the host here. The password will be used for Secure Shell connections and AutoLogins.

Also, if you connect to the host you can send the password by defining a macro key as `^&`. For example, if your F12 key is set to send `^&`, you can answer the password prompt of the host by pressing F12 (see Options→Session Profile→Macro Keys).

However, in most cases it will be more convenient to let the AutoLogin feature (see below) handle the entire login.

SSH Key

You can select a specific SSH identity file (private key) for login with the SSH hosts.

If, instead of a private key file, you provide the full name and path of a PKCS#11 compatible DLL, ZOC will use the PKCS#11 DLL to access your smart-card and retrieve the identity from the SmartCard chip.

It is also possible to choose a key from the Windows Certificate Store (this includes keys from windows compatible hardware, e.g. CAC/PIV readers or most usb based keys).

It is also possible to use standard identity files for the login (these standard authentication files are configured via Tools-menu→SSH Global Authentication Keys).

Note: SSH Authentication is only available if you enter a username for login.

Login Script/Parameter

The name of a REXX script to be executed after ZOC has made a connection to the given host. An optional argument can also be supplied, which will be passed to the script and which the REXX script can access in the form `loginarg= ARG(1)`

Such a script can be used to log you into the host automatically and/or to perform more complex automated tasks after being logged in. However, for simple login procedures (sending user name and password), the AutoLogin feature (see below) is usually sufficient.

If both an AutoLogin procedure and a REXX script are given, ZOC will first log in using the AutoLogin procedure and will then start the REXX script.

AutoLogin Sequence

This field contains a sequence of events which automatically logs you on to the host. The events are either Wait=<something> or Send=<something>. Since most logins can be handled in a form like 'wait for this, then send that', ZOC offers this simple method to perform automatic logins without having to deal with the more complex REXX programming.

Note: This method is not necessary for SSH connections, because SSH offers a standard procedure to exchange user credentials. To send special keys like Enter or ESC in an event, you can use codes from the the control codes table. The most common control codes are `^[` for Esc, `^M` for Enter, `^&` for the password and `^%` for the username (assuming that username and password were provided in the upper part of this dialog).

Example:

```
Wait=Select Host:
Send=Server^M
Wait=login:
Send=^%^M
Wait=password:
Send=^&^M
```

What this sample does: Wait until the remote host sends `Select Host:` then simulate keystrokes for `Server` and the Enter key (`^M`). Then wait for the host to send `login:` and answer by sending the username field from this dialog and the Enter key. Having done that, wait for the `password:` prompt and send the password from this dialog to the host.

Record keystrokes on next login

It is not necessary to write the step by step AutoLogin procedure by hand. To let ZOC record the sequence of events, please enable this option and enter the username and password which you are going to use for the login into the fields in the upper part of the dialog and save the entry. Then connect to it from the host directory main dialog and log into the host manually. After you are logged in, select Stop Recording from the Script menu to let ZOC fill the AutoLogin field for you.

1.5.1.4 FTP

This settings page allows you to set up an FTP server which is associated with the host of this host directory entry. E.g. you can set up an SFTP connection with your Linux secure shell login, or an FTP server that corresponds with your zOS host.

The FTP client will allow you to exchange files with that server more easily (e.g. up- and download files with drag drop instead of using SCP or IND$FILE commands).

FTP Type

This field allows you to choose the type of FTP connection offered by the server (e.g., SFTP (SSH) or FTPS (SSL)) for the internal FTP window, which is part of ZOC Terminal.

Alternatively, you can choose to use an external FTP client (e.g., FileZilla of WinSCP). The external FTP client itself is configured under Options→Program Settings→Special Files. Parameters for invoking the external FTP client to connect to this specific server can be provided below.

Use a different server for ftp-connections

Check this option if the ftp server associated with this connection has a different address than the server you connect to for terminal access.

Use different login for ftp authentication

If the ftp server uses authentication credentials which differ from the username and password in your host directory entry, you can check this option and provide the FTP username and password here.

Local/Remote start folder

These are folder designations which will be used as starting points when the FTP windows opens. The local folder is a folder name on your computer (e.g. `C:\users\you\Desktop`) and the remote folder is a folder name based on the remote system (e.g. `/home/you/development`).

Parameter for ext. FTP

If you choose an FTP type of "External" above, you can enter a parameter here, which will be passed on to the external FTP client to specify where to connect. Typically, this will be a URL-like string containing the username, password, and server name, e.g., `sftp://harry:secret@hogwarts.edu`.

The parameter can contain one or more of the following placeholders which will be replaced with the respective values from the host directory entry: `%USERNAME%`, `%PASSWORD%`, `%FTPSERVER%`.

The Set button next to this field will populate it with a typical sftp-URL parameter containing placeholders which will be suitable for FileZilla and WinSCP.

1.5.1.5 Shortcuts

Add this host directory entry to the main window's File menu

Select this option, if you want the host to be accessible from ZOC's File menu.
You can insert a &-character before a character in the name for the host directory entry (on the main page, e.g. `&ZOC InfoBBS`), to create a shortcut for the menu item.

Connect via F-Key

It is also possible to define a Ctrl+F-Key (e.g. Ctrl+F6) combination that can be used to connect to this host.

Create Desktop Shortcut ...

This button creates an icon on your desktop which will start ZOC and automatically connects to this host. The button will also let you create an entry in the user buttons of the current session profile.

1.5.1.6 Window

The Window settings for each host control if the host is called in its own terminal window, in a new tab or if the existing window and tab should be used.

If you want to open connections in new tabs, you should configure the entry to Plain Connect (see Host configuration in this dialog). The reason is that when launching multiple connections at the same time, ZOC will sometimes need to queue the connect requests if the entries are set to connect via AutoConnect (see the description for the Connect function in the Host Directory window for details).

If a new window is selected, you can control the position where on screen the new window should be opened. The position is given in screen pixels and there is a button to take the value of the current window. However, when opening the new window, the width and height of the window may vary, because it will also be influenced by the font and window settings in the session profile.

1.5.1.7 BBS

Call to this host is due every ... days

If ZOC finds a carrier detect signal after calling a host, it will update the date of last call for the called entries (and all others that are equal to the one that was called in the first six characters of the name, thus managing multi-line hosts correctly, see Name-field above).

If you want to call a host regularly (say once a week) you can enter the number of days here after which ZOC should remind you (by showing a symbol in front of the phone number in the main window) that it is time to call that host again.

IEMSI Options

Many bulletin board systems offer a method called IEMSI to automate the login procedure.

Using this method you can provide your name in the IEMSI options window and set the password field to the password you use to log into that host. Additionally you can specify some options that are used by host.

During the login the host will send the IEMSI request (**IEMSI_IRQ; not to be mixed up with the **EMSI_REQ that is used for FIDO mailer software) to which ZOC responds by sending your user name and password to the host. If everything goes well, the host will let you in at once and show (or skip) news, new mail and new files as selected in the IEMSI options window.

1.5.1.8 Info

Notes about this Host Connection

This first line of the memo stores a short note about the host. You might want to enter the name of the sysop or other info about the host here (for passwords you should use the AutoLogin feature or the password field).

If this field is not empty, it will be displayed in the host directory main window. If you do not like this, begin the memo with a space character; in this case, the main window will still show the number of calls instead of the memo.

The other fields are maintained by ZOC and cannot be modified.

1.5.2 Host Directory Settings

Visible columns

Depending on your preference you can disable the display of the secondary columns in the host directory.

Remember selected entry when showing list

This option lets you configure the host directory to open with the entry preselected that was marked when the host directory was closed.

Show tool tips for host directory entries

When you hover the mouse over a host directory entry, a small window (tool tip) with information about the entry will appear. This option lets you turn off the information window.

1.5.3 Create Shortcut

This function lets you create an icon on the desktop or a button in the current Session Profile's user

buttons to initiate the connection with a single click.

1.5.4 The AutoConnect Feature

When you selected one or more entries from the host directory or choose Quick Connection from the File Menu the AutoConnect window will pop up. It will try to connect you successfully to the number(s) you selected. If a number is busy it will try the next in the list (if more than one was selected) or retry the number. To do that, AutoConnect will use the values you set in Options→Program Settings→AutoConnect options to define the maximum number of retries and the time between attempts to call the same number.

During AutoConnect you can control the operation using four buttons:

Cancel
Abort calling.

Retry Now
If AutoConnect waits to call the next number you can skip the delay and retry at once.

Skip This
This function skips the call to a number and proceeds with the next in the list. The skipped number will be called later.

Drop This
This function cancels the attempt to call a number and proceeds with the next in the list. The cancelled number will be removed from the connection list.

A special feature is that the window switches between a small and a large version if you click on the minimize/maximize button in the upper right corner of the window.

1.5.5 Importing other Host Directories

Import from other Terminal Emulators

ZOC is able to read and convert host directories of a few popular comm packages (like Putty, Telix for DOS and Windows, Telemate for DOS and Windows, etc.). To do this, select the type of host directory, the section of the ZOC host directory to which you want to import the entries and the file to import. Then press the OK button.

CSV/Text Import

To import host directory record from a different format (database or spreadsheet), you need to create a text file which is suitable for CSV/Text import. The text file needs to have one entry per line with columns separated by a specific character (e.g. comma, semicolon, vertical bar, etc.). Most database or spreadsheet software is able to export a file in CSV format, which should be suitable for text import. The file needs to contain at least columns for the Name/Label field and for the Connect to field with optional columns for username, password and port in any order.

Example:

```
Router 1;192.168.1.1;Markus;asdfg77
Router 2;192.168.2.1;Tom;qwert88
Router 3;192.168.3.1;Kate;uiop99
```

Custom Imports

If you have data which goes beyond what the import functions are offering, you can also create a file in ZOC's own format. The ZOC host directory (HostDirectory.zhd inside the ZOC data folder accessible from the File menu) is a plain text file. Using a scripting language like Visual Basic, Perl, REXX, it should be possible to create entries in that format you see inside the host directory file without too much trouble.

Post Import Adjustment

If you want to adjust multiple entries after import, you can mark multiple records in the host directory and click Edit to modify them in one operation. This can be useful to change a settings in all of them to the same value, e.g. setting them all to Xterm emulation, etc.

1.6 Other Functions

- ❑ User Defined Actions
- ❑ File Transfer Window
- ❑ IND$FILE Transfer
- ❑ SCP Transfer
- ❑ Zmodem Transfer
- ❑ Datastream Viewer Window

1.6.1 User Defined Actions

User buttons, keyboard keys, auto-macros, idle action, and other events can be configured to perform functions such as sending text, initiating a connection, opening a document, etc. This dialog lets you choose actions and providing additional details about how to perform it.

Many of these choices are straightforward, but here are more details for the ones that are not:

Send text to remote computer

The text to send can include control codes from the list shown in List of Control Codes. For example, the text can contain `^M` for Enter or `^I` for Tab or `^Z` for the Enter/Submit button in a TN3270 emulation.

(Note: For emulation keys that have no control code equivalent you can use the 'Send a mixture of text and emulation-specific keys' action instead).

Send a mixture of text and emulation-specific keys

This function is useful in cases where it is necessary to send combinations of text and special emulation keys that do not have a control code equivalent.

The parameter to this action is text to send, but it can also contain the name of emulation-specific keys which can appear in angled brackets. For example, in this action `Hello world!<Enter>` is the equivalent of `Hello world!^M` in the Send text to remote computer action, but this format is more flexible because you can use all the available named emulation keys, as in `<Home>submit<F3>save<Enter>`.

Note: if you need to send actual < and EXMP(>) signs, you can use `<lt>` and `<gt>` as placeholders for these, e.g. `diff -u file1.dat file2.dat <gt>files.diff<Enter>` will send `diff -u file1.dat file2.dat >files.diff` followed by the Enter key.

Send a single emulation-specific key

This action sends a specific emulation key, such as the VT220 Do key or the TN3270 Insert key. It is mainly intended for simple cases where you merely want to map an emulation function, like the TN3270 PA1 key, to a key or key combination on your local keyboard. You can find the names of the keys for each emulation by clicking the '&threedots;' button next to the key entry field or in the

Names of Emulation Keys for User-Defined Actions section of this help text.

(Note: To send multiple emulation keys or a combination of normal text and emulation keys, use the Send a mixture of text and emulation-specific keys action instead).

Perform a function from ZOC's main menu

It is also possible to perform the function from the main menu (e.g., to map it onto a button or different key combination). This is achieved by entering the name of the menu function into the action field or by selecting the function by clicking the '...'-button.

Connect to a host using a host directory entry

To connection to hosts which are stored in the host directory, specify the name of the host directory entry or choose it from the host directory via the '...'-button.

Connect to a given address via specific communication method

This action initiates a connection that is not listed in the host directory. The parameter is in the format `<connection type>!<destination>`, where the connection type is one of the types shown in the session profile while the destination depends on the connection type (e.g., for a Secure Shell connection, it will be an IP address or host name and port, for a Unix Pipe it would take the form a a file name, etc.). Examples are `Secure Shell!joe@nowhere.com`, `Secure Shell!harry:secret123@hogwarts.com:10022`, `Telnet!3270.mainframe.com:10023`, `Serial/Direct!COM5`, etc.

Run a REXX script in the current tab

To start a REXX script from a key or button you will enter the name of the script file in the action field, e.g., `script\myscript.zrx` and click on the button that tells the assistant that the text in the field is the name of REXX script to run.

1.6.2 File Transfer Window

The transfer window consists of the following parts:

Name

The name of the file to be transferred (not available in Xmodem downloads).

Size

The size of the file to be transferred (not available in Xmodem downloads).

Transmitted

The number of bytes sent or received so far (the net value without control bytes like checksums).

Time

The duration of the transfer so far and the projected time to go (only available if the file size is available).

Speed

The average net number of characters sent or received so far. This should be about a tenth of the connect speed (e.g., 960 cps at 9600 bps).

Skip

This button is available in Zmodem only and skips the current file.

Note: The skip option may not work with all Zmodem implementations.

Disconnect after transfer

If set, ZOC will disconnect from the remote host after the file transfer

Note: Please be careful, ZOC will not properly logoff from your host, but it will merely hangup instead. Not all hosts (or sysops) like this.

Delete file after transfer

This checkmark makes ZOC delete the file after the transfer. Files are only deleted if they were transferred without error and file transfer was not aborted by the user.

1.6.3 SCP File Transfer

The ZOC SCP file transfer is an interactive variant of the SSH `scp` command.

This dialog allows downloads of single files, multiple files (using file name patterns), folders and folder trees.

Important: SCP transfers are always performed in the context of the initial SSH session. Using the su command or jumping to another server with a remote ssh or similar command will possibly make the SCP transfer fail.

By default the operation will be performed on the user's home folder (not the current working folder), except when you specify a path to a different folder in the source field. By default the source field will be preset to your current remote working directory which ZOC obtains by issuing a pwd command on the remote server.

The source field will be interpreted as a single file name or file name pattern (e.g., `~/Documents/*.txt`), unless you click the recursive option.

If recursive is activated, source needs to designate a folder (e.g., `~/Project/`) and all files from that folder and all subfolders will be downloaded.

Downloaded files will be stored in the destination folder, which by default is ZOC's standard download folder (the standard folder is configured in Options→Program Settings→Folders). The destination field can be edited directly but it needs to point to an existing folder on the local computer.

If incoming filenames match the file extensions to store in the alternate download folder (configured in Options→Session Profile→Transfer), the files will be moved to the alternate download folder instead (configured in Options→Program Settings→Folders).

1.6.4 Zmodem File Transfer

Overview

Most Unix/Linux systems haven an implementation of Zmodem installed and preconfigured (if not, look for a package named lrzsz.zip on the internet, check if the Linux distribution offers a rzsz or lrzsz packet or ask our support). This program works very well to transfer files over SSH or telnet connection and integrates seamlessly with ZOC.

If rz/sz is available on the remote server (or lsz/rsz in some implementations), you can send a file from the remote host to ZOC by typing `sz <filename(s)>` on the remote shell. ZOC will then open the Zmodem download dialog and place the file into the local download directory (Options→Program Settings→Folders).

Transfer

To send a file to the remote machine, start ZOC's upload (Transfer menu). ZOC will then first send an `rz` command to the host to start the remote receiver and will then perform an upload. Alternately you can type `rz` on the remote shell and ZOC will open the upload file selection dialog.

Options

For file transfers using ZModem, ZOC offers a variety of options to control the transfer. These can be configured in the session profile (transfer section):

Upload File Management

These options apply only to uploads. They are only effective, if the receiver is able and willing to handle them, so they may or may not have an effect on an actual file transfer, depending on the receiver's Zmodem implementation (they work best when used to send files to Omen Technology's RZ or DSZ, which is available on various platforms including Unix, VMS and MS-DOS).
You can specify that you want the receiver to convert carriage returns and line feed according to its platform and you can suggest what the receiver should do if a file already exists on its machine. Options are to use the receiver's default action, to always overwrite files, to overwrite files if their file date is older, to resume an aborted file transfer or append the data to the existing files (the corresponding Unix SZ options are -p, -y, -n, -r, -+).

ASCII

The ASCII options are used when it is necessary to do an end-of-line character translation between different operating systems. It is useful when transferring text file, but has disastrous results when applying to binary data like ZIP files or executables.

Ignore time stamp

Select this option, if you want received files to be marked with the date/time of download, instead of being marked with the time stamp the file carried on the file system of the sender.

1.6.5 IND$FILE File Transfer

The IND$FILE file transfer protocol is intended for use with the TN3270 emulation. It allows the exchange of files with an IBM zSeries host.

While connected to a host via TN3270 you can choose Upload or Download from the Transfer menu in order to send files to or receive files from the host.

When selecting one of these functions, a dialog will pop up and ask for the local and remote filenames. Local filenames can be any file directly available to your computer. The remote file is the name of a dataset which the host can understand (e.g., a sequential or partitioned dataset name on a TSO/ISPF system like `SOURCE.ASM(TEST01)` or `'TXUSER.OUTPUT.LOG'` (including the single quotes for full qualification).

The transfer options govern the conversion of the data between the host and local computer, e.g., insertion of CR/LF line endings, conversion between EBCDIC and ASCII or remapping of the ASCII data into the currently selected TN3270 codepage.

The DCB option allows you to add extra parameters to the IND$FILE command, e.g., `LRECL` or `RECFM` parameters. See TSO-Receive and TSO-Send at
https://www3.rocketsoftware.com/bluezone/help/v42/en/bz/DISPLAY/IND$FILE/IND$FILE_Technical_Reference.htm

The setting for the host type modifies the command which is sent to the host in order to initiate the transfer. TSO is intended for a TSO prompt (genuine TSO READY prompt or for the command prompt on ISPF menu 6), ISPF/TSO can launch a file transfer from any ISPF Command ==> or Option ==> prompt while VMS is (obviously) intended for VMS systems.

1.6.6 Datastream Viewer

The datastream viewer window allows you to scroll through the text you have received since starting ZOC (or since clearing the capture buffer). You can also use it to view debugging and data-trace output, which

can be configured under Session Profile→Trace/Debug.

You can mark text with the mouse (thus putting it into the clipboard) or search for text by pressing the F(ind) key.

1.7 Customizing ZOC

There are various types of options that allow the customization of ZOC on different levels: Program Options, Misc. Program Options and Session Options.
Finally some customization can also occur through the commandline.ini file.

Program options hardly change once you defined them. On the other hand, it is very likely that some other will differ between different hosts you use (these are called session profiles). Therefore, while there is only one set of program options, but you can have as many Session Profiles (containing options that are more likely to be session specific) as you like.

Session Profiles, Key Map Profiles, etc. are files which contain configurations, which may differ between connections. E.g. when connecting to a Linux web server and an IBM mainframe, you may want to have different sets of user buttons, different screen layout and cursor types. A session profile is a set of configuration parameters which can be assigned to a connection (session).

Additionally, when making connections through the Quick Connect dialog and Host Directory, you can specify a connection type and emulation and the options for these. These will override the corresponding settings in the session profile. This provides a layered set of settings, that will let you use the same session profile (f-keys, user buttons, screen elements, etc.) on hosts which use slightly different connections and emulations.

See also: Options Menu and Session Profiles, Keyboard Profiles/Translation Profiles, Program Settings.

1.7.1 Program Settings

Program options (as opposed to the options configured in Session Profiles) are used or all connections. They contain options which, once they are set, typically hardly ever change.

Program settings are

- Window
- Tabs
- Mouse
- Clipboard
- Scrollback
- Local Typing
- Printer
- Sounds
- Prompts
- Stored Passwords
- Folders
- Special Files
- Quick Connect
- AutoConnect
- Disconnect Action
- Miscellaneous

- Features
- Updates
- Master Password
- Language

See also: Options Overview, Options Menu, Configuring ZOC, Session Profiles, Key Map and Character Translation Profiles.

1.7.1.1 Window

Window Title

This field can contains the text that will be displayed in ZOC's window title bar. The text can contain placeholders which will be replaced with values depending on the current situation. e.g. `ZOC %VERSION% %CONNECTEDTOHOST% %OPTIONS%` or alternately `%COMPUTERNAME% %CONNECTEDTOHOST% %OPTIONS%`.

%CONNECTEDTOHOST%

The string "connected to "and the name of the remote host.

%COMPUTERNAME%

The name of your computer.

%HOST%

The name of the remote host (e.g. the name of the host directory entry).

%HOSTRAW%

The connect-to field (e.g. ip or hostname).

%INSTANCE%

The number of the currently open ZOC window.

%OPTIONS%

The name of the currently loaded session options file.

%VERSION%

The version number of ZOC.

%ZOCORHOST%

If not online the String "ZOC" or, if online, the name of the remote host.

%ZOCVERSORHOST%

Like `%ZOCORHOST%` but including the version number.

Maximized window border color

When the window is maximizes, there will be a area that fills the gap between the terminal area and the window borders (this happens because there may be size restrictions for the layout of the terminal, see Session Profile→Layout). Normally this area is drawn in a light gray, this options changes this to another color.

Extra border around terminal

This options draws a border around the terminal area to create a visual gap between the window

border and the text. The color of the border is the same as the background color from the session profile.

Window Transparency
Setting this option to a value other than zero will make the program window partially transparent. Please be aware however, that with older graphics boards, this **may add considerable strain on the CPU**.

See also: Advanced Window Settings, Program Settings

1.7.1.2 Advanced Window Settings

Hide window while connecting via AutoConnect feature
If you do not want the large ZOC window on screen while using the AutoConnect (dial and retry) feature, you can select this option. In this case ZOC will hide while attempting to make the connection, just leaving the small dial progress window on the screen. The main window will come back after a connection was made (or finally failed).

Bring window to front when connected
If set, the main window of ZOC will come to the top of all windows when a connection was made. This way you can have ZOC connect in the background and jump in if a connection attempt was successful.

Minimize window during file transfers
If you do not want the large ZOC window to be visible while a file transfer is going on, this options minimizes the ZOC's main window when starting the transfer so that only the transfer progress window remains on screen.

Bring window to front after a file transfer completed
If set, the main window of ZOC will come to the top of the desktop after a file transfer has been completed. This way you can put ZOC aside to do other work during a file transfer without missing the transfer's end.

Default fallback font
This option specifies ZOC's default font, which will be used whenever ZOC is unable to open a specific font. That situation can occur for example, if you are sharing ZOC configuration files (e.g. session profiles) among users on different computers or when transferring them between the Windows and macOS version of ZOC.

See also: Window, Program Settings

1.7.1.3 Tabs

Action when opening new tab ...
The action which is configured here, will be performed when *additional* tabs are opened through user interactions (e.g. from the menu). You can choose to display the Quick Connect or Address Book window or you can run the default startup REXX script `Startup.zrx` (the script will be passed a parameter of `##NEWTAB##` to indicate that it was triggered through a new tab).

Show close button for tabs
This option lets you define, which of the session tabs will show a close button.

Show tabs always
When more than one session is opened within a ZOC Window, the program will show a bar with tabs

for each session. This option controls if the tabs bar will remain visible when only one session is open.

Access tabs via ...
This option lets you activate key combinations to quickly switch to the first nine tabs. Under Windows the key combination is Alt+digit, on macOS it is Cmd+digit.

Rename tabs when receiving a Xterm terminal-title sequence
Remote hosts can send a command to display a text in the window title. When this option is active, that command will also rename the title of the tab in which it was received.

Mouse wheel switches tabs when hovering over tabs bar
If the mouse pointer is over the area of the tabs bar, scrolling the mouse wheel will switch forwards/backwards between tabs.

Colorful Tabs
If this option is active, the color from a tab's color marker will be used to tint the background of the tab, the background of the user button bar, the chatline and the statusbar. This helps to make switching between sessions visually more distinct.

When making a connection from the host directory, a tab's color will be determined from the host directory entry's icon color. This way a connection to a host from the host directory will always have the same color.

When this option is active, but when no tabs are shown (i.e. when there is only a single session), the window elements are only tinted if the current connection has been assigned a specific color from the host directory.

See also: Program Settings

1.7.1.4 Mouse

Mouse buttons
Please pick a choice from the lists to map a functions to the right or left mouse button.

Note: Some of the choices, those that are related to marking/copying depend on the option Automatically copy selected text to clipboard in Program Settings→Clipboard. The Send Enter choice, when invoked in a TN3270 and TN5250 session will first move the cursor to the clicked location, thus, for example, allowing the selection of ISPF menu using the mouse.

Mouse wheel
This option offers a list of choices for the action that the mouse wheel performs. These are mostly obvious, except the Smart mode: In smart mode, the wheel enters scrollback except when a remote application (e.g. Linux VIM, screen, MC, etc.) has switched the terminal to alternate-screen mode, where the mouse wheel will move the cursor. Essentially this means that the mouse wheel will enter scrollback while you are in a Linux shell, but it will move the cursor while you are in a screen based application.

Word delimiters
When double clicking on screen to mark a word, the list of word-delimiters characters will determine which characters will are considered part of a word and which will stop the mark.

See also: Program Settings

1.7.1.5 Clipboard

Automatically copy selected text to clipboard
When you select text in the terminal area, with this option is enabled, the program automatically copies the selected text to the clipboard.

Clear selection after copy
Selecting this option will clear the selection of text in the terminal area, after the selected text is copied to the clipboard.

Suppress 'Text was copied to clipboard' notification
This option can be used to suppress the small confirmation window that ZOC usually displays when copying text to the clipboard.

Use Ctrl+C and Ctrl+V for copy and paste
In terminal software, Ctrl+C and Ctrl+V are key combinations which send specific codes to the remote computer (e.g. Ctrl+C is usually sending a hex 03 byte which some system use to interrupt an ongoing procedure). If you do not need these functions, you can set this option to use Ctrl+V to paste text from the clipboard into the session and Ctrl+C to copy selected text to the clipboard.

Note: Ctrl+C is only affected while text selected on screen. With no selected text, Ctrl+C will send hex 03, which is mostly interpreted as a 'break'.

Use Ctrl+Shift+C and Ctrl+Shift+V for copy and paste
Same as above, but you also have to press Shift in addition to Ctrl. So the native Ctrl+C and Ctrl+V codes will be sent as key codes, but when used together with the Shift key the clipboard functionality can be accessed.

Invert Alt-key behavior for text selection
Normally text selection will be done in flow mode. Pressing the Alt-key while marking text will change the selection mode to block mode. This option will toggle this behavior.

Immediately paste single-line selections
Selecting this option will always send marked text immediately, if only one line was marked on the screen. If the option is half-selected, the text can still be sent immediately if you hold the Shift key pressed while doing a single-line selection.

Trim trailing blanks from last line in selection
This option deletes trailing blanks from the last line of any text selection.

Trim trailing blanks from all lines in block selection
This option deletes trailing blanks from every line, which often happens during block selection.

See also: Program Settings, Edit Menu

1.7.1.6 Scrollback

Scrollback Viewer
In this group of options you can specify the maximum number of lines which the program will store for scrolling back. Very large numbers may consume a lot of memory and will come with a slight performance penalty.

There is also an option to include alternate screens in scrollback. Linux programs like VI, Midnight Commander write to an alternate screen buffer and their data normally clutters the scrollback. The notable exception to such programs is a tool called screen. If you are using screen, you may want to

enable this option to get the output from screen included in the scrollback.

Some hosts/programs clear the screen using a series of commands that delete single lines. If you want these to be included in the scrollback, check the corresponding option. However, if you find the scrollback cluttered with single lines that appear there out of sequence, you may want to disable the option.

From ZOC's menu you can either view the scroll back contents within the main window (via Shift+PageUp) or within an extra window (via Alt+PageUp). For the latter function (scrollback in window) you can also select a font.

Data Stream Viewer

The data stream buffer logs incoming data exactly in the order it is received to a memory buffer and it has a viewer that displays the data that way. The data stream viewer can also show data in debug mode (see Session Profile→Trace/Debug). The size of this buffer can be configured in this dialog.

The window for the data stream viewer will use the same font as specified above for the scrollback viewer window.

See also: Program Settings, View Menu

1.7.1.7 Local Typing

Enable the interpretation of control codes

If this option is enabled, control codes like ^M will be resolved (a return will be sent). Otherwise they will be sent as text (the text ^M will be sent).

Select Font

Here you can select the font name and size, which is used in the local typing entry field.

See also: Program Settings

1.7.1.8 Printer

Printer

Here you can either select the printer to which output is sent when you use one of the printing functions. Under Windows it is also possible to select the Windows default printer or a printer port.

If you are printing from a remote application (e.g. via vt100/220), the appropriate printer depends on the application. If the *application merely sends text* (without specific printer control codes), you can either select any printer that is installed in your system.

However, if your application *does send printer control codes* (requires transparent printing) you either need to select a printer port (LPTx) or you need to select the printer type that your remote application expects and activate the Bypass printer driver option below.

Bypass printer driver

When this option is set, remote print data will be sent to the printer directly without reinterpreting the data. This is necessary if the remote application is sending printer control codes to do page formatting or print forms. In this case the type of your physical printer needs to match the type, which your remote application is expecting, because neither ZOC nor the operating system will adjust printer data between the two.

Raw print settings

If necessary, you can specify character sequences which will be sent to the printer, prior to and after sending the actual print data.

These sequences can contain ZOC control characters (e.g. `^(1B)` or `^[` to send an Esc character) and they can be used to send a printer specific code, for example, to switch the printer to a different paper tray.

To-File print settings

If you choose a printer destination of To-File, you can specify a file name that will receive the print data. The name can contain placeholder codes to insert time, date or session title into the name.

The post-processor field allows you to specify a command that will be called once the print job is complete. You can use the placeholder `%SPOOLFILE%` or `$(SPOOLFILE)` to insert the name of the spool file into the command, e.g. `"C:\WINDOWS\system32\notepad.exe" "%SPOOLFILE%"` or `/usr/bin/open -t "$(SPOOLFILE)"`.

Print job merging

When printing to the local terminal printer (via VT102 codes) some programs send start/stop printing codes for every line, causing the Windows spooler to print every line onto a single page.

This option controls the timeout (in seconds) which ZOC uses to merge subsequent printing jobs into one (0 disables this feature).

Page header

The page header field lets you define data which will be printed on top of each page. The header line consists of parts for the left, center and right of the headline, separated by vertical bars, e.g. *<left-part>|<center-part>|<right-part>*

For each of these parts you can use the following macros, which will be replaced with their corresponding values: *%USERNAME%*, *%COMPUTERNAME%*, *%DATE%*, *%TIME%*, *%PAGE%*, *%CONNECTEDTO%*, *%TABNAME%*.

For example, use `%COMPUTERNAME%|%USERNAME%|Page %PAGE%` to print the computer's name on the left, the currently logged on user's name in the center and the Text "Page nn" on the right.

Note: This feature will be ignored if you enable the Bypass printer driver option above.

See also: Program Settings

1.7.1.9 Sounds

Sound Files

These are sounds to be played to indicate various events in the program.

Leave the fields empty or enter a non-existing name to play no sound.

Remote Beep (Bell)

Most remote hosts use an acoustic signal to draw the user's attention to errors or other special conditions. You can configure the program to represent that signal by playing a beep sound from the speaker, playing the default system sound, playing a custom sound file or by displaying visual effects.

Additionally you can specify a maximum number of such signals within a given time. If the system sends more bell signals than the given threshold, additional signals will be suppressed until the

same time passes without receiving any more bell signals.

See also: Program Settings

1.7.1.10 Prompts

Define if you want to be warned before performing "dangerous" operations or if you want to protect the setting of the macro keys and custom buttons.

Confirm before ...
Select if you want to be prompted ('Are you sure?') to confirm various actions throughout the program.

Show warning when modem is missing (no CTS signal)
The CTS signal on the serial port normally indicates the presence of a device attached to the port. If you are using hardware that does not offer a CTS signal, you can disable this option in order to avoid the warning about non-present devices.

Note: This option only applies to the Serial/Modem and Serial/Direct communication methods.

Show warning for file transfer via modem without RTS/CTS handshake
The RTS/CTS handshaking method provides reliable control of the data flow between the PC and the modem. If you use serial communication with higher transfer speeds without RTS/CTS handshake, you will very likely encounter transmission errors. Enabling this option will give you a warning if you select more than 4800 bps without RTS/CTS.

Note: This option only applies to the Serial/Modem and Serial/Direct communication methods.

Safety ...
These options can make sure that current activity is not inadvertently interrupted by closing the program.

See also: Program Settings

1.7.1.11 Stored Passwords

Stored SSH Passwords
These options refer to passwords which were stored by using the Remember password checkbox in various password and passphrase dialogs.

SSH Passwords Expiration
You can set an expiration to delete passwords which were not used for some time or you can delete some the existing passwords manually by means of the Manage Passwords button.

See also: Program Settings

1.7.1.12 Folders

Scripts
The folder where REXX files are searched by default and where the several REXX file requesters start.

Profiles
The directory where the host directory, session profiles, key maps, translation tables, etc. are stored or loaded from.

SSH Files
This folder is the default location for SSH related files (public/private key file, known hosts lists, tunnel profiles, etc).

Download
The directory into which files are downloaded.

Alt. Download
This is a folder, into which special files are downloaded. The file extensions of the files to be put into this directory are defined in the Options→Session Profile→File Handling dialog. This directory is mostly used to redirect special file types to an alternate directory (e.g. pictures or media).

Upload
The folder from which uploaded files are taken if no explicit path is supplied in the file name.

Text Upload
The default location from which the Transfer→Send Text File and Transfer→Send Binary File function take their files.

Logging
The logging feature stores the logs here (see Options→Session Profile→Log).

See also: Program Settings

1.7.1.13 Special Files

Editor
This options defines the editor to be used when calling Edit function (e.g. Edit Rexx Script). If you set this field to the string `<DEFAULT>`, Notepad.exe (Windows) or the default editor (macOS) will be used.

FTP-Client
This field specifies an external ftp client to be used when a host directory entry is configured to use the external ftp-client rather than the built-in ftp window. It is intended to be a fully qualified path to an executable, e.g. `C:\Program Files\FileZilla FTP Client\filezilla.exe` or `/Applications/FileZilla/Contents/macOS/filezilla`. This executable will then be invoked with a host specific parameter which can be defined in the FTP tab of each host directory entry.

Shell
Shell defines a shell to be used, when calling the Command Shell function in the View menu. This shell is also called with the /C parameter if you issue a ZocShell command from within a ZOC REXX script.

Note: If you set this field to `<DEFAULT>` ZOC will use CMD.EXE for Windows and /bin/sh for macOS.

Connection Log

This file is a file in which ZOC logs connections and file transfers (C+/C- in the log means connect/disconnect, DL means download, UL means upload).

The filename can contain placeholders like `Calls^2^3.log` for date/time (see placeholder codes in the appendix).

If no path is specified, the call log is stored in the log directory. If the field is empty, no calls log is written.

Save Screen

This field defines the name of the file, into which the File Menu→Save-Screen function will save the contents of the terminal window.

ZOC-Events

The REXX script selected here will be called, when ZOC is started, when it terminates and when another communication method (connection type) is activated. You will find the technical details inside the sample file `ZOCEVENT.ZRX` and in `ZOCEVENT.TXT` in the script folder.

File Transfer

Here you can select which REXX script is called before a file transfer begins and after a file transfer was completed. The REXX script can modify the file name before the file is actually written to disk.

This can be used if you want to implement a complex file storage scheme (e.g. if ZIP files are uploaded from different directories than other files or if you have special download directories for ZIP files, mail files and pictures) or if you want to automatically run an file compression utility before uploading certain files (and upload the compressed file instead). Additionally you could automatically unpack all downloaded ZIP files or decrypt incoming PGP files etc.

You will find the technical details inside the sample file `ZOCXFER.ZRX` and `ZOCXFER.TXT` in the script directory.

Alternate REXX DLL

By default ZOC uses an internal REXX processor (Regina REXX) which is provided together with the ZOC program.

However, if you prefer a different REXX language implementation, or if you are already using a different REXX product that is compatible to the IBM's original REXX API, you can specify one or more DLL names here (multiple entries are separated by a vertical bar).

The following is a list of entries to be made to use the most common commercial and freeware implementations:

ZOC-REXX (current ZOC default)

`zocrexx.dll` (ZOC-REXX is a variant of REGINA REXX, see Copyrights).

Enterprise REXX (ZOC v5 default)

`RxREXX.dll` (found in the program folder of ZOC5)

Regina-REXX

`regina.dll` (http://regina-rexx.sourceforge.net)

Object-REXX/OOREXX

`rexxapi.dll|rexx.dll` (http://www.oorexx..org)

Quercus Personal REXX

`WRexx32.dll` (http://www.quercus-sys.com/prexx.htm)

Reginald-REXX

`reginald.dll` (http://www.borg.com/~jglatt/rexx/reginald/reginald.htm)

Important: Please note that you need to provide full path names to your installed DLL, e.g.
`c:\Programm Files\oorexx\rexxapi.dll|c:\Programm Files\oorexx\rexx.dll`

See also: Program Settings

1.7.1.14 Quick Connect

The settings below refer to the File-menu→Quick Connect window.

Do not preselect last connection in Quick Connect dialog
This option affects the Quick Connect window. When set, the field containing the destination address for the connection will not be preset with the value from the last connection.

Number of entries in history
This settings determines how many entries from recent connections are stored in the list of connections in the Quick Connect window.

Clear history
This button will clear the list of recent connections in the Quick Connect window.

1.7.1.15 AutoConnect

The AutoConnect function is used when you select the connect function from the Host Directory. It only applies to host directory items, that have the AutoConnect feature selected. This feature is mainly intended for modem/ISDN connections.

The feature tries to connect you to a given host and possibly retries several times if it fails to make a connection. You can specify the maximum number of retries and the time between trials of the **same** phone number. If, for example, you set the time between attempts to 10 seconds and select three entries from the host directory to be called, and if the call to all of them fails, the first one will be called immediately after the third since trying the second and the third will very likely need more 10 seconds.

Some countries require some time to pass before the next number can be dialed. So ZOC lets you define the minimum time between tries to call **any** number.

Additionally you can define a string to be used as a prefix for calls via Modem/Serial or ISDN connections. This is useful if you have to dial 0 or 9 to get a dial tone. If you need the prefix only for the Modem/Serial handler, you can use the modem dial commands (see Modem Options) or the phone number macros (see below) for this purpose.

Small AutoConnect window

You can choose between a large and a small window that shows the progress and details while calling. The large window (naturally) covers more screen while the smaller one might not fully display long messages or phone numbers (which is just a cosmetic problem).

Note: You can switch between the larger and the smaller window while calling by clicking the minimize/maximize buttons in the upper right corner of the AutoConnect window.

Dial Macros

The host directory offers four special macros for use with telephone numbers. When you use the strings $1, $2, $3 or $4 somewhere in a telephone number (or other device's equivalent) ZOC replaces the string with the text defined in the Macros dialog.

E.g. if you are using calling card numbers for some entries in your host directory, you could define $1 as

1234 5678 9012 3456 and insert $1 into your phone number where ever the calling card number should appear.

Note: These macros are strictly for use with telephone numbers in the phone book. They do not work if you use them in modem strings, function keys, etc.

Note: In many cases similar results can be achieved by the use of different dialing strings (see Modem Options).

See also: AutoConnect Device Responses, Program Settings

1.7.1.16 AutoConnect Device Responses

Here you provide ZOC with the responses, which the modem and other connection handlers return when making or failing to make a connection.

These are used by ZOC's AutoConnect feature to determine if a connection was made, if ZOC should retry a failed connection (e.g. if a modem connection was busy) to or drop a called host directory entry or if the connection attempt has finally failed (e.g. because of an incorrect hostname or cabling problem).

The responses need to be separated by a vertical bar (|) without space characters between them like `CARRIER|CONNECT`.

See also: AutoConnect, Program Settings

1.7.1.17 Disconnect Action

Clear screen
This options will clear the screen and reset the screen colors after you disconnected.

Show host directory
If this option is enabled, ZOC will show host directory after you disconnected from an online service (only if there are no more entries left in the call queue).

Load standard session profile
If this option is enabled ZOC loads the default option set after you disconnect from a host session.

Call next queued entry without prompting
If you disconnect from a service and if there is another item in the call queue (assuming you had marked more than one entry to connect to in the host directory), ZOC can either try to connect to the next service automatically or after prompting you if you want it to.

Close tab
This option automatically closes a session tab when the connection ends (this only applies when multiple tabs are open).

Close window
This option automatically closes the ZOC window when a connection ends.

Pressing the Enter-key after a disconnect tries to reconnect
When this option is enabled, pressing the Enter-key in a disconnected session will try to reconnect to server.

See also: Program Settings

1.7.1.18 Miscellaneous

Show cursor position in status line as 1-based

When the display of the cursor position is enabled in Options→Session Profile→Window Parts, this option determines if the position of the top left corner is shown as 0/0 or 1/1.

Always open Rexx file selection dialog in default Script folder

Usually the file selection box for REXX scripts opens in the folder and with the file name of the last recently used REXX script. With this option turned on, the file selection will always start in the script folder, which is configured in Program Settings→Folders.

See also: Program Settings

1.7.1.19 Features

Communication and Emulation Priority

This block of options groups various connection types and emulations in order to tell the program loader how likely these are to be used.

Off means that it is unlikely that you will use the functions. On means that the function is one of your primarily used functions. The third state (neither on nor off) is somewhere in between (meaning that the feature will be used occasionally).

Depending on these settings, ZOC may delay the loading of functions and may not offer them in the Quick Connection or other dialogs unless they have already been used. Please be aware that turning off a function off does not mean that it is strictly disabled. Think of it in terms of telling the likeliness of using the function in order to optimize loading the program's loading time and memory use.

ZOC may still load and offer such a function, if it has previously been used. If for example you have disabled the TN3270 emulation, but subsequently have a host directory entry, which has the TN3270 emulation selected, it will be still be loaded when you open the host directory.

See also: Program Settings

1.7.1.20 Updates

Update Settings

Here you can change how often ZOC connects to the EmTec website at program start to see if there is an program update available.

See also: Program Settings

1.7.1.21 Master Password

The ZOC master password or certificate is used to encrypt sensitive data (e.g. passwords stored in the host directory or remembered SSH passwords). It can also be used to limit access to some functions of the ZOC software.

You can specify your own password. The password is case sensitive and you will need to enter it every time it is required for an restricted/encrypted operation.

Alternately, under Windows, you can select a key from a cryptographic certificate (e.g. smart card, usb-key, Yubikey, etc.). This needs to be a certificate that includes an RSA or ECC key which resides on CAC/PIV compliant hardware that support cryptographic signing oder encryption functions. Typically this will be something like a Yubikey with a self signed key in a PIV slot of the Yubikey. Alternately you can

create a self signed certificate and key and transfer it to a device like an eToken.

If you lose the password or the cryptographic key, the encrypted information will be permanently lost.

A password is required for the following operations

Select various parts of the program that should be protected by a password, because they are likely to contain confidential information.

There is also an option to lock the program if there was no keyboard input for some time. This works across multiple ZOC windows, i.e. the program will lock only you did not use the keyboard with any of the open ZOC windows.

Likewise, the password to protect the program window only needs to be entered for the first ZOC window and will not be requested if another ZOC window is also open and locked.

1.7.1.22 Language

Language Setting

Here you can change the application language. If you choose 'Automatic', this means that ZOC will choose its language based on Windows display language setting. Otherwise the chosen language will be used.

See also: Program Settings

1.7.2 Keyboard, Character Translation, etc.

Global program options (as opposed to Session Profiles) are valid for all connections.

Such program global options are Program Settings and Toolbar Icons.

Additionally, ZOC offers Keyboard Mapping Profiles and Character Translation Profiles. You can create multiple versions of these profiles, which then can then be mapped to different host directory entries.

See also: Options Overview, Options Menu, Configuring ZOC, Session Profiles, Program Settings.

1.7.2.1 Remapping the Keyboard

Basic Remapping

The keyboard mapping function (Options→Keyboard Profiles) allows you to remap all keys on the keyboard using your own text, special characters or ZOC functions.

This function takes precedence over all other keyboard processing in the program. If you remap keys like backspace or Alt+D you will lose the key's default meaning (i.e. you will lose backspacing depending on the emulation or the shortcut to open the host directory).

To remap a key, select Options→Keyboard Profiles. Then set/unset the checkmarks for the key qualifiers (Shift, Ctrl, Alt, Num-Lock, Scroll-Lock) which are associated with your redefinition. E.g. to remap `Ctr+I` set the checkmark for Ctrl, and leave the other qualifiers off or unspecified.

Then click on the actual key to bring up the key redefinition window. (Alternately, in many cases it is possible to just press the key combination that you want to redefine.)

Scroll/Num Qualifiers

Please be aware that ZOC treats the Num-Lock and Scroll-Lock state as key qualifiers. This means that you can remap keys differently, depending on Num-Lock, Scroll-Lock or both. If you want to map a key independent of the Num-Lock or Scroll-Lock state, set the check marks for these qualifiers to unspecified (the checkboxes for these qualifiers have three different states: on, off, unspecified).

It is even possible to map text to the Num-Lock and Scroll-Lock keys themselves. If you do this, these keys will still toggle the corresponding LEDs on your keyboard and hence their state. In that case it is important that all your other keyboard mappings are made with the Num-Lock and Scroll-Lock checkmarks set to 'unspecified'.

Other Functions

The Show/Print buttons will display a list all your key definitions including all the qualifiers that apply to them.

Using the Load and Save-As functions, you can create and manage multiple key maps which can be assigned to host directory entries in the host directory's edit dialog.

See also:
Options→Session Profile→F-Key Macros, Keyboard Topics, Emulation Dependent Key Names, Keys for the VT102/VT220 Emulation, Keys for the 3270 Emulation, Keys for the 5250 Emulation

1.7.2.2 Redefining Single Keys

After you selected a key to be redefined in the keyboard redefinition window, a window will appear, which lets you define a user defined action that is performed, when you press that key together with the selected combination of key qualifiers (e.g. Alt+Ctrl+Z).

For details about the available types of actions and their specific function and parameters, see User Defined Actions.

Getting Key Mappings from other sources: If you found mappings key on the internet or in a program which you previously used, or if you extracted such mappings via the `infocmp` command from a Linux system, the following table shows some common ways in which these are expressed in other sources and how to convert these into a format for use ZOC' 'Send Text' action (see below for a few examples).:

Other Software	ZOC Equivalents
^M \r CR Return	`^M` `^(0D)`
Esc \E \e \033 \x1B	`^[` `^(1B)`
^[[	`^[[` `^(1B)[`

CSI	`^[[`
	`^(1B)[`
	`^(9B)`

SS3	`^[O`
	`^(1B)O`
	`^(8F)`

For example if you issue the `infocmp` command on a Linux system using the Xterm emulation, you will

for example see an entry kf12=\e[24~, which defines the mapping for the F12 key (type `man terminfo` on a Unix shell for a list of all the cryptic key and sequence tags like kf12).
According the table above, the ZOC equivalent for \e[24~ will be `^[[24~` or `^(1B)[24~`

Note: The full list of supported control codes by ZOC can be found here: Control Codes and Action Codes

1.7.2.3 Translation of Inbound/Outbound Characters

Overview
Users in countries that do not use the US-Ascii character set, often end up with the problem that the codes of characters that are sent by the host do not match the codes used by the terminal program and/or operating system.
Using the Load and Save-As functions, you can create and manage multiple translation tables which can be assigned to host directory entries in the host directory's edit dialog.

Note: The translations are only active if the option to ignore them is not enabled in the Options→Session Profile→Terminal.

Note: Before building a translation table, you should check if the character codes used by your host are already supported as a character set, see Options→Session Profile, Layout).

Case Study
Let us assume your host sends (and needs to receive) the character `x` instead of the German letter `Ä` (as some hosts in Germany do) and that you have a keyboard that has a letter `Ä` key.
Let us also assume that ZOC is configured for the IBM/DOS character set (see, Options, Session Profile, Window). In this case ZOC generates the code 142 when you press the `Ä` key on the keyboard and will expect the code 142 to be received to display an Ä in the terminal window. (If ZOC is configured for Win/ANSI or running in VT220 mode, the same would apply to the code 196.)
However, instead of the expected 142 the host sends the code 123 (which ZOC knows as x).
Hence, to make the two systems interact correctly, you will map the code 123 in the receive table to 142. To do this double click line 123 in the left part of the window and enter 142 in the field for the decimal code and click ok. `123 x -> 142 Ä` will appear in the window. This will take care of codes that are received from the host. However, to correctly convert the code that ZOC sends when you press **Ä**, you will have to make an inverse mapping in the right table (double click 142 and change it into 123).

Here is a list of the common Umlauts and the code that ZOC uses, depending on the current configuration (Character Set setting in Window Options).

Character	DOS/IBM	ANSI/Windows (aka Latin-1)
Ä	142 (8 E)	196 (C4)
ä	132 (84)	228 (E4)
Ö	153 (99)	214 (D6)
ö	148 (94)	246 (E6)
Ü	154 (9A)	220 (DC)
ü	129 (81)	252 (FC)
ß	225 (E1)	223 (DF)

1.7.2.4 Setting up the Toolbar

The toolbar makes functions easily accessible by offering icons for them.
To customize the toolbar you can select/deselect icons from a list by clicking on them with the mouse. Entries displayed on the right side will be shown in the toolbar (if the window is wide enough).
Additionally (depending on screen resolution and your personal taste) you can select if you want the icons in the toolbar to be large or small and if you want them to be packed tightly or not.

1.7.3 Session Profiles

A session profile is a predefined set of options (terminal, pane, color, etc.) that can be used as the underlying configuration for a connection. You can have multiple session profiles, and each can be used for one or more connections. This has the advantage that you do not have to configure the same settings over and over for each host.
The Save and Save As buttons are used to store the configuration into a file which can later be assigned to a session in the Host Directory or in the Quick Connection dialog.
Available tabs are:

- Connection Type
- Terminal
- Layout/Font
- Colors
- Cursor
- Window Parts
- Emulations
- Text Sending
- Logging
- User Buttons
- Keyboard
- Auto-Action
- Auto-Highlight
- File Transfer
- File Handling
- Idle Action
- Trace/Debug
- Advanced

See also: Options Overview, Customizing ZOC, Options Menu, Keyboard and Translation Profiles, Program Settings.

1.7.3.1 Connection Type

This window is used to select and configure the available connection methods. It can be opened via Options menu or by clicking on the leftmost button in the status bar.

Connection Type
This list defines the standard operation parameters for all connection methods. It also selects the default connection method for this session profile.
However, the selections and settings made here may be ignored later if the connection (e.g. a host directory entry) to which the session profile is assigned overrides them.
For example, if the session profile is assigned to a Host Directory Entry, the entry itself can be set to use a different connection type. This entry can then also be configured to override the session profile's options for that particular connection type (e.g. the session profile may have the keep-alive option disabled for telnet connections, but a host directory entry can override the telnet options to change this behavior for that particular connection, which is initiated by this entry).
In other situations, e.g. when connections are made from the commandline or if the profile is activated through REXX scripts, the settings will be used exactly as they are defined here.

Help for the individual methods is available in the Connection Type section of this help text and you will find details and descriptions of the option for the specific types in the following list:

- Secure Shell: Overview and Options
- Telnet: Overview and Options
- Telnet/SSL: Overview and Options
- Serial/Modem: Overview and Options
- Local Shell: Overview and Options
- Windows-Modems: Overview and Options
- Named Pipe: Overview and Options
- Rlogin: Overview and Options
- UDP: UDP (Datagram) Overview and Options.

See also: Session Profiles, Connection Types, Customizing ZOC and Options Menu

1.7.3.2 Terminal

This window is used to configure general terminal and text output characteristics.

TERMINAL-OPTIONS

Local echo

Normally a remote host will send all typed characters back to the terminal, which then displays them. In other words, the users sees his own input by way of the host echoing the characters back to the terminal. If a host does not do this (i.e. if the user does not see what he types), the terminal can use this option to display input locally.

Don't CR/LF for local echo

When the Enter key is pressed and when local echo is also enabled, this option makes ZOC echo only a Carriage-Return instead of the normal CR/LF combination.

Incoming end of line -> CR/LF

When enabled, this options ensures that each incoming line end (CR or LF) is treated as an Carriage-Return/Line-Feed combination. Use this option if received text is either printed over and over in the same line without advancing to the next line or if the text is printed like a stairway with the cursor moving down at the end of the line but not moving back to the left edge.

Soft Shift-Lock for letters

When using remote systems that require (or accept) only upper case characters, this option can be used to treat all letter keys as if Shift-Lock was active. Non-letter keys (e.g. digits or punctuation) will remain unaffected.

Scroll jumps

Even in modern computers, scrolling large amounts of text can be a demanding task for the CPU. In order to improve scrolling speed, you can allow ZOC to delays scrolling until a few lines of text have arrived. This way ZOC saves CPU cycles, because needs to move the text only once instead of five times.

Obviously output speed benefits from this (while the scrolling appears rather "jumpy" instead of

smooth). The bigger the jumps you select (normally flea jumps should be enough) the more lines ZOC will collect before actually scrolling and the faster output will be.

ADVANCED

Answer to ENQ

Some hosts use the ENQ character (^E, hex 05) to request information from the terminal. This option lets you specify the answer which will be sent, if an ENQ character is received from the host. If the string contains the placeholder `%USERNAME%` or `%COMPUTERNAME%` or `%DOMAINNAME%` (alternately `$(USERNAME)`, `$(COMPUTERNAME)`, `$(DOMAINRNAME)`), these will be replaced by their respective values.

Disable URL highlighting

When this option is active, URLs (http links, ftp links, etc.) on screen are not highlighted when the mouse hovers over them and a click on an URL will not open a web browser.

Send echo

Loop back all incoming characters to the originator. This option can be used if other computers are connecting to ZOC and if they are expecting it to behave like a host computer.

Warning: This may result in an endless loop if the originator also sends an echo, which is true for most modems and hosts.

Strip high bit from incoming data

If enabled, all characters in the terminal window are printed with 7 bits. This can be used if you receive graphical characters where text should appear (of course, this does not help if bursts of noise from the phone line send garbage to your screen).

Software 7E1

This option is used if a host with 7E1 coding is accessed through an 8 bit Gateway (ISDN, Telnet).

Ignore remote beep

This option lets you suppress remote beep signals (^G/Bell). It overrides the setting in Options→Program Settings→Sounds.

Ignore character translation profile

This options can disable the translation of characters via Options→Character Translation Profiles. If enabled, a loaded character translation profile will remain dormant.

See also: Session Profiles, Customizing ZOC and Options Menu

1.7.3.3 Layout/Font

Please also read the topic Fonts, Window Size, Color for additional information regarding this dialog.

FONT AND TERMINAL SIZING

... window size will determine number of rows/cols

When the ZOC window is sized, this option will keep the font size unchanged, but change the

number of rows/columns of the terminal window instead (e.g. from 80x24 to 87x32).

If this option is activated, you will not be able to manually select a specific terminal size (rows/columns), because the terminal size will be determined by the size of the window.

... adjust font size proportionally

When you change the window size, this option will keep the number of rows/columns fixed and the program will instead display the terminal area using a larger or smaller font.

If this option is activated, you will not be able to manually select a specific font size in this dialog (see below), because the font size will be automatically determined depending on the size of the window. Also, sizing of the window with the mouse will only work in larger steps because only those sizes will be offered, for which there is a matching font and the natural font proportions (height to width ratio) will be retained.

... adjust font with/height (stretch font to window)

This option is similar to the proportional font sizing (see above), but allows the program to disregard the natural font proportions and stretch or condense the font width in order to adjust the terminal area to the window size. This will allow a greater variety of possible window sizes but may result in letters and characters looking less natural.

... retain font size and rows/cols

This option will keep the font and logical terminal size. This will prevent the ZOC window from being changed in size.

TERMINAL SIZE

Number of Columns/Rows

You can select the terminal size here (columns and rows). If in doubt, try terminal sizes of 80x24 or 80x25.

Note: If an emulation or host requires a specific terminal size (e.g. fixed size TN3270 or VT100 in 132 columns mode), this setting may be ignored.

FONT AND CHARACTER DISPLAY

Character Set

ZOC for Windows supports various character sets, e.g. the DOS/IBM character set, Windows/Ansi/Latin-1 character set and the Linux coding for Unicode (UTF8).

The choice of character set depends mostly on the host and is required for correct display of line graphics or accented characters.

Terminal Font

Select a font size for the terminal window from the list. The font list will only show fonts, where all characters have the same width (non proportional fonts). This is required for correct display of tables etc.

The font size is given in display pixels width x height (this is different from the "points" size, which is usually used in text processing programs). The size of the terminal area (where text is displayed) depends on the terminal size (number lines and characters per line) combined with

the font size.

Antialiasing

With this option enabled, the font will be rendered on screen with smoother edges, although with small fonts this can result in a slightly fuzzy look. Turning the option off will create a more crisp but slightly edgy font. The third state of the option (half checked) will activate or deactivate this setting based on the operating system's default.

See also: Fonts, Window Size, Color, Session Profiles, Customizing ZOC and Options Menu

1.7.3.4 Colors

This window is used to select and configure the color settings of a ZOC Terminal session.
Color settings are organized into two tiers:
Color Selection: Most colors are selected from a predefined palette. For instance, you can choose to display marked text in orange from the active palette.
Palette Customization: The actual color value used depends on the chosen palette. To fine-tune the colors, you will need to edit an existing palette or create a new one.

Note: If you are using TN3270 emulation, be aware that its color handling differs significantly from most other emulations. Consequently, TN3270 provides an additional color configuration option. You can access this through Session Profile→Emulation→TN3270. This option utilizes some colors from the standard configuration window and also uses the color palettes defined here.

COLOR SETTINGS

Colors

Here you can choose from a predefined palette of colors for the fore and background and for rendering characters with specific attributes, e.g. you can display characters in yellow when the host wants to display them underlined. Depending on the terminal emulation, the remote host can change the display colors to highlight text.

ZOC's default is black text on white background. If you encounter odd color effects with your host, please try the classic gray text on black background.

The colors of the first 16 entries in each palette are industry standard, although their individual shades may vary slightly. These colors should be changed with caution. If you still wish to modify them, you can use the Edit Color Palette button to assign new colors to each entry. When a remote host requests one of the base colors (e.g., "green"), ZOC will use the color you defined in the global table.

TN3270/5250 Note: The 3270 and 5250 emulations technically require complete color sets. Hence the foreground and background colors and some of the attribute related colors like highlight, underline, etc. will be ignored. Instead ZOC offers a selection of color schemes (combinations of some of the 16 base colors) for these 3270/5250 emulations. These can be selected in Session Profile→Emulation. Selection of the color palettes and modification of the global color table (see below) will affect the colors in these TN3270-configurations.

Palettes

IIn this section, you can select from various color palettes for the base colors. While most palettes are based on the same set of colors, they have slight variations in hues for colors like light blue, light red, green, etc.. You may prefer to choose a palette that you are accustomed to. Additionally, there are palettes such as Solarized that are not based on the typical 16 ANSI colors but on a different color system.

See also: Fonts, Window Size, Color, Session Profiles, Customizing ZOC and Options Menu

1.7.3.5 Cursor

CURSOR APPEARANCE

Format
Form of the cursor, e.g. Block, Vertical or Underline.

Mode

Blinking cursor
This option makes the cursor in the main window blink. Another setting on this page will also allow you to adjust the blink frequency.

Crosshair
When this option is selected, thin vertical and horizontal lines are drawn on the screen to indicate the cursor position.

1.7.3.6 Emulations

This window is used to select and configure the emulation. It can be opened via menu or by clicking on the emulation button in the status line. The selected emulation and its options will be used as long as no other part of the program overrides it.
For example, if the session profile is assigned to a Host Directory Entry, the entry itself can yet activate a different emulation. The entry can also choose to use or to override the session profile's options for that particular emulation.
Specific keys of each emulation can be mapped to different PC-keys. This is done by using Emulation Specific Key Names in conjunction with Options→Keyboard Profiles.

Emulation

Xterm
Todays most widely used emulation in the Unix/Linux world. The emulation supports most of the available pc-keys as well as color, international characters and it also offers a couple of other advanced features. It should be chosen whenever possible. On the host side it should be known as TERM=xterm

See also Options for the VT/Xterm/ANSI Emulations.

ATT4410
The ATT4410 is essentially a VT220 emulation, where only a few keys send different codes. It is mainly used to access and configure AT&T switchboards.

ANSI BBS
The ANSI Emulation is mostly used in BBSs. It is main feature is that it uses the IBM-PC graphic character set and allows to display text in different colors. For compatibility with the FIDO ANSI-BBS

standard, it also supports all VT102 control codes. Under Unix it is commonly referred to as TERM=pcansi

See also Options for the VT/Xterm/ANSI Emulations.

ANSI SCO

A slightly different implementation that uses different codes for the extended keys (F-keys, Ins, End, etc.). Unix systems that support this emulation usually know it as TERM=scoansi

AVATAR/0+

An emulation mainly used on the ancient FIDO BBS network.

TN3270

The 3270 family of emulations is used with IBM mainframes. The key definitions are listed in 3270 keyboard mapping. ZOC does support 3270 extended color coding and also offers a range of predefined color styles for non-extended sessions. You can also find session profile named (OPTIONS\Settings_3270.zoc) that provides user buttons for some common and some unmapped keys.

Please see the description of the individual TN3270 emulation options.

TN5250

The 5250 emulation is used with IBM iSeries eServer (former AS/400). The emulation needs to be used with Telnet connections (this combination is commonly known as TN5250). The key definitions are described in 5250 keyboard mapping. You can also map other special keys to other places on the keyboard or to user buttons by using emulation dependent key names.

Linux

Your choice of emulation when dealing with Linux hosts. The emulation supports most of the available pc-keys as well as on screen colors. On the host side set TERM=linux

QNX

Also available is an emulation for use with the QNX 4.2x real time operating system. This emulation was originally developed by Ingenieurbuero Jurk in Weisenheim and Mr. Jurk was kind enough to provide us with the source code for general inclusion with ZOC.

Sun CDE

This emulation is the preferred emulation on Sun Solaris (Common Desktop Environment) machines. It is based on VT220 but offers color support together with f-keys and extended keys. Set your Solaris machine to TERM=dtterm

TTY

This emulates a teletype style output. All characters are printed and no further functions exist, except CR, LF and TAB. In a way, this is a non-emulation.

Televideo

This is a group of older terminal emulations (TVI9xx). TVI terminals are is still used in some specific areas, but they should only be chosen, if your host explicitly requires any of them.

VT100/VT102

VT102 is mostly used in Unix environments and is compatible with the (slightly inferior) VT100 standard. VT102 is a simple all purpose choice. It does not support colors or extended keys except F1-F4, so you should try to use vt220 instead. Under Unix it is known as TERM=vt100 or TERM=vt102

See also Options for the VT/Xterm/ANSI Emulations.

VT220

VT220 Terminals are a major improvement over VT102 and a good starting point if you not know what emulation to choose.

VT220 supports most of the keys found on a PC keyboard (please check VTxxx keyboard mapping) and it supports national character sets in a consistent way. However there is no official support for colors. Almost all Unix systems should support his emulation if you set TERM=vt220 on the shell, although most systems today are optimized for the xterm emulation.

See also Options for the VT/Xterm/ANSI Emulations.

VT420/VT520
These terminals are further improvements over the VT220 although most of the improvements are not really relevant to Unix/Linux shells. They will be the emulation of choice for VAX/OpenVMS systems though.

See also Options for the VT/Xterm/ANSI Emulations.

VT52
VT52 is a very old standard from DEC and is hardly used anymore. It is not upward compatible with VT100, VT102 and VT220.

Wyse
This is an ancient terminal emulation, but it is still used in some areas. It should only be chosen, if your host explicitly requires it.

TERM=
With Telnet and SSH connection, ZOC will report your current emulation to the remote host, which usually will store it in the TERM= environment variable for use with termcaps/terminfo applications. All ZOC emulations have a built in TERM string (as indicated in the above list) that will work on almost all such systems. However, if you need Telnet or SSH to report a custom term type identifier to the remote host, you can enter it here. If you are not sure what this means, please leave this option off.

See also: ENQ option in Session Profile→Terminal

See also: Session Profiles, Customizing ZOC and Options Menu

1.7.3.7 Options for the VT/Xterm/ANSI Emulations

List of Options for the VT/ANSI/Linux/Xterm Emulations
Below is a list of non trivial options, which in some or multiple emulations from the VT/Xterm/Linux/ANSI family (options which should be obvious or self-explanatory were omitted).

Options

Enable UTF8 (Unicode) Support
VT and their derived emulations originally only supported 8-bit character sets similar to Latin-1. This option changes the behavior to interpret 8-bit characters as UTF8, which is the common standard for Linux hosts these days.

Use Alt as Meta key
Some Unix applications (like Emacs or MC) are using the Alt-Key combinations to access special functions. Enabling this option makes Alt+Letter and Alt+Digit combinations send the pressed key prefixed by an ESC character.

Note: When you enable this option, you will not be able to access ZOC's menu shortcuts.

Map PF1-PF4 to top row on keypad
The original DEC terminals had the PF1 to PF4 keys located above the numeric keypad. With this

options ZOC uses the key pads Num, /, * and - keys as PF1 - PF4.

Send navigation keys with Alt/Shift/Ctrl qualifiers
This option affects the Xterm emulation and the keys Home, End, Ins, Del, Page-Up, Page-Down and the cursor keys. When the option is active, these keys will be sent to the remote host if they are pressed in combination with Alt, Ctrl and/or Shift. However, if the option is active, this will override some shortcuts within the ZOC menu (e.g. Ctrl+Page-Down for Download).

Disable xterm Extensions
xterm adds a range of control codes to perform specific operations not native to the original VT-emulation (e.g. changing the program title or certain display features). ZOC VT-emulations support these, but allows to disable them.

Disable ANSI-like Colors
The original VT-emulation were monochrome and did not support colors. ZOC VT-emulations supports color support with control codes, which were borrowed from the ANSI terminals. If necessary these can be disabled.

Disable application keypad mode
In VT102 mode the numeric key pad is used as a replacement for VT102 auxiliary keypad (see VT102-keyboard). This can be problematic if you are using the cursor keys in the numeric keypad instead of the gray cursor keys. Enable this option, if you want to use the cursor keys in the numeric keypad and if your remote application does not use the VT102 keypad.

See VT Keyboard for a description of the keypad.

Backspace sends ^H instead of DEL
When enabled, the backspace key sends the ^H code when backspace was pressed (otherwise ^?/7F is sent). The Del key remains unaffected.

Del key sends ^D
This option causes the Del key to send Ctrl+D instead ^?/7F. This is useful on Unix systems like ibm-uss (IBM z/OS UNIX System Services), which do not have a default mapping for the Del key.

Swap Backspace key and Del key
If you enable this option, a backspace (^H) is sent when the DEL (^?/7F) key is pressed and vice versa.

Destructive backspace
This options erases characters from the screen if a backspace is received. Otherwise the cursor merely moves back one step.

Alternate F-Keys
Typically on hardware VT terminals, the F5 keys has a local meaning and is not transmitted to the host. This option allows to shift all F-keys to the left, sending F5 as F6, F6 as F7 etc..

1.7.3.8 Options for the TN3270 Emulation

The TN3270 emulation has a range of options specific to this emulation. Those with a non-obvious meaning are explained here: **Options**

Swap Ctrl and Enter Keys
The normal mapping for these keys is that Enter acts as the NewLine key and the right Ctrl key acts as Transmit. Enabling this option swaps these keys. (See also: Default Key-Mappings for the 3270 Emulation).

Use shift+arrow-keys to mark text on screen
Text can be marked on screen using the shift key. Otherwise the shift key will move the cursor

sideways at double speed.

Make Fnn and PFnn fields clickable on screen
This options will highlight areas on screen that refer to F-keys (e.g. "F3=End") and allows you to click these fields in order to send the corresponding F-key.

Move cursor when pasting
This option will move the cursor when pasting text from the clipboard, otherwise the cursor will remain at the position where the pasting began.

Allow pasting of matching content over protected fields
If this option is set, you can paste text over protected areas if that pasted text matches the protected content. This allows you to copy a whole screen with contents in the fields and paste it back into the same screen later (filling the fields again). If this option is disabled, when pasting reaches a protected area, it will jump to the next field and will continue to paste the content there (this is similar to PCOMM's behavior).

Answer SF-Query OEM-AuxDev requests
When ZOC receives an SF-Query request, it will include the OEM-Aux Dev 8F(hex) section into the reply, which includes information about the terminal emulator, the session name, etc.

Colors

Color Settings for Standard Mode
You can choose from predefined sets of colors that represent screen attributes (e.g. protected, unprotected, highlight, ...). These apply only to TN3270 sessions in standard mode, i.e. when ZOC is not showing "-E" after the model number in the blue status line.

Color Settings for Extended Mode
This dialog lets you create your own color schemes for TN3270 sessions that run in extended mode. (when ZOC is shows an "E" after the model number in the blue status bar). In extended TN3270 sessions the host requests specific colors for text displayed on the screen. The color map lets you replace the host's choices with a different color, e.g. when the host requests blue in a TN3279-E session, you can map that color to purple.

Background and Statusbar Colors
In these sections you can specify the colors for background and statusbar. These apply for both, standard mode and extended mode.

1.7.3.9 Text Sending

The options below are used for the functions Send Text File, Send Binary File and for pasting from the clipboard.

Delay per character/Delay at end of line
Sets the time (in milliseconds) to wait after sending a character and after each line. This is useful if the host cannot keep up with fast input (e.g. in online editors).

Note: The line delay is sent in addition to the character delay. If the character delay is zero or if you are sending a binary file, then the line delay will be ignored.

End of line character
Sets the form, in which end of line characters are transmitted to the remote machine. When set to AsIs, characters will be sent as they appear in the file, otherwise either only CR or LF or CR/LF is sent.

Note: End of line translation is ignored when sending binary files.

Note: emulations (TN3270, TN5250).

See also: Session Profiles, Customizing ZOC and Options Menu

1.7.3.10 Logging

This window is used to define file logging parameters.

Logfile name

The name defined here will be used as filename for the logfile. The name can contain absolute or relative folder paths (folders in the path, especially when containing placeholders, will be created as necessary).

It is also possible to create dynamic names and folders by using placeholders for date and time, which will be substituted with their respective values when the log is opened (see placeholder codes in the appendix). Examples are `^+_^1^2^3.log` or `Host_^+\Log_^3_^2_^1.log` (the latter is a logfile in a dynamically created subfolder).

Note: Unless you supply an absolute path with the logfile name, the file will be placed in the log directory which is configured in Options→Program Settings→Folders.

If logfile already exists ...

If a logfile already exists, it can either be overwritten, or the new data can be appended to the existing logfile. As an alternative the filename can be incremented by an appended number to make sure that no existing logfile will be overwritten.

Logging is now active

Enable saving incoming data in the logfile. This function is identical to setting checkmark near the logfile name in the status bar or to using the Log to File entry in the Logging menu.

Create a new logfile when the date or hour placeholders change

If this option is active and if the filename for the logfile contains placeholders for year, month, day or hour, ZOC will create a new logfile as soon as a new day/hour/etc. begins.

Add time markers to each line

This option will insert time stamps before each line of the log. The time stamps will indicate, when exactly the data was received.

Log session headers

Session headers are a few lines of extra information in the log. They contain information about when a session starts, which host you connected to etc.

Discard emulation control codes from log

Cursor placement, colors and other screen controls for terminal emulations are performed by adding invisible control codes to the data. This option controls, if these codes should be added to the logfile or not (normally they are of little interest to the end user).

DC2/DC4 control

Some hosts are able to remotely suspend/resume logging by sending DC2 (^R) and DC4 (^T) characters (the checkmark in the status line will turn gray if the hosts suspends logging for you). This option enables/disables the ability of the host to control the logging.

See also: Session Profiles, Customizing ZOC and Options Menu

1.7.3.11 Window Parts

Please also read the topic Fonts, Window Size, Color for additional information regarding this dialog.

WINDOW ELEMENTS

Menu

This setting lets you hide the menu-bar below the title bar (Windows only). When hidden, you can access the menu functions by right-clicking the terminal area.

Toolbar

This option controls the visibility of the toolbar.

User buttons

Enable or disable the area containing User Buttons (below the toolbar).

Status line

Here you can select, if you want to see the status line.

Status lights (LEDs)

If the status line is enabled, you can select to show status lights to indicate the connect state, states of the program (e.g. to indicate if a REXX script is running) and data send/receive activity.

Traffic indicator

If the status lights (LEDs) are turned off, ZOC will show an icon inside the leftmost button instead. This icon indicates if a connection is active or not (icon in color or b/w).

This icon can also indicate if data is currently sent or received. Depending on your preference, you can disable the traffic indication (in that case it will only show the online/offline state).

Logfile name and logging status

ZOC offers a function to log the data traffic into a logfile. In the status bar, there is a field that shows the filename and indicates whether logging is currently active or not. Note: This option has three possible states: 'On', 'Off', and 'Visible only when logfile is active'.

Show cursor position instead of rows/columns

This option will display the current cursor position in the status bar where normally the window size is shown. In mixed mode (half checked) the window size is shown, when the cursor is in the upper left corner of the screen, otherwise the cursor position is displayed. The display of the top left can be switched from 0/0 to 1/1 using an option in Options→Program Settings→Miscellaneous.

Vertical scrollbar

Here you can turn off the scrollbar at the right of ZOC's main window.

Snippets

ZOC monitors your input data stream for filenames, web and email addresses and collects them in a small window that floats near the ZOC window (see View Menu).

This option can hide (grayed) or show (checked) the window or turn this feature off (unchecked) to save CPU load.

See also: Session Profiles, Customizing ZOC and Options Menu

1.7.3.12 User Buttons

Below the tool bar resides a another bar containing user defined buttons. The user buttons can be managed in this session profile section.

User Buttons

The user buttons, when clicked, can perform a variety of user define functions, e.g. the sending of text or commands to the remote host, the simulation of a ZOC menu command or the execution of a REXX script. Here you can add, modify or delete user buttons. You can also group them in folders or import the whole button set from another session profile.

You will find more information about the functions that can be mapped to individual buttons, by editing a button and then clicking the Help button in the User Defined Action window.

See also: Session Profiles, User Defined Action, Customizing ZOC and Options Menu

1.7.3.13 Keyboard

F-Keys

This dialog lets you map text or special actions to the top row of your keyboard. It is limited to F-Keys, but a more versatile keyboard mapping feature can be found in Options→Keyboard Profiles.

Using this feature, it is possible to make F-keys send text, initiate connections from the host directory, execute REXX scripts, etc. You will find more information about these functions, by editing a F-key and then clicking the Help button in the User Defined Action window.

Note: Be aware though, that mapping functions to F-keys here will override the emulation function of these keys. If they have a defined function in an emulation (e.g. Xterm, VT220 and Linux have predefined values for all f-keys, VT102 uses F1-F4, etc.), the user setting will be performed instead.

AutoMacros

In addition to the F-key macros, ZOC also provides a feature which we call Auto-Macros.

You can think of AutoMacros as abbreviations. Whenever ZOC finds that you typed some text that is found in the abbreviations, ZOC will replace it with the full text you provide here or performs the associated function. This is done by sending backspace characters to delete the abbreviation (if necessary) and then running the defined action.

To avoid accidental triggering, abbreviations are matched case sensitive and replaced only if there is a non alphanumeric character typed before and after the text. So, if you defined `Com` as an abbreviation it will be replaced if you type `Data-Com_` or `Com-Port`, but not if you type `The Com1-Port` or `DataCom_`.

It is still recommended to append a period or an exclamation mark to the abbreviation text (like `MS.` or `lo!`) to make it unique and to avoid erroneous invocation.

This feature is similar to Auto-Action and some results can be achieved by using both methods. However the important difference is: AutoActions monitor the incoming data stream (sent by the host), AutoMacros watch the keystrokes from local the keyboard.

Note: This features is not available if you are using block oriented emulations (TN3270, TN5250).

Note: This feature is disabled during REXX execution and during AutoLogin and Learn mode.

SCROLL-LOCK KEY ACTION

This option lets you map events to the scroll lock key.

Ignore Scroll-Lock key

ZOC ignores the scroll lock state of the keyboard.

Activate local typing

Pressing the Scroll Lock key toggles the chat field (see Window options and Screen Elements).

Toggle doorway mode

Pressing the Scroll Lock key toggles the keyboard Doorway mode. Doorway mode is an terminal feature which sends DOS-like key codes for special keys like F-Keys, PgUp, etc. This is required by some arcane control software like RemoteBBS or OS2YOU.

See also: Session Profiles, Customizing ZOC and Options Menu

1.7.3.14 Auto-Action

AutoActions

Auto action provide a way to automatically send text or invoke ZOC actions whenever a sequence of specific characters is received from the host (see User Defined Action).
This feature can be used to

- ❑ Automate logins: You could automatically send your username and password when receiving the text `Username:` and `Password:`
- ❑ Reading mail: When you receive the `You have mail` notification from your Unix host, you could start your mail reader by automatically replying `elm^M`.
- ❑ Skip birthday checks: Some hosts ask for your date of birth now and then to make sure no one else uses your account. Using AutoActions you can provide the date automatically when receiving the prompt.

This feature is similar to AutoMacros and some results can be achieved by using both methods. However the important difference is: AutoActions monitor the incoming data stream (sent by the host) while AutoMacros watch the keystrokes from local the keyboard.

Warning: You should make sure that the text you are looking for is unique. Making the trigger text too general increases the risk, that your actions will be performed at undue moments.

Note: This feature is disabled during REXX script execution and during AutoLogin and Learn mode.

Note: emulations (TN3270, TN5250).

See also: Session Profiles, Customizing ZOC and Options Menu

1.7.3.15 Auto-Highlight

The auto highlight feature allows you to highlight specific phrases (e.g. an important system prompt) on screen, using a different color for text or background or both.
You can add, edit or delete such phrases and assign or change the given colors (choosing 'Standard' for a color means that no special color will be chosen for fore- or background).

Plain Text:
The phrases will be matched to the screen output as they appear. However, they can be set to ignore upper/lowercase and they can be set to contain a limited set of placeholders.

Placeholders:
When you enable placeholders, the trigger text can contain characters that do not represent themselves, but which instead represent a certain class of characters: A `#` (number sign) will match any digit (0-9), a `@` (at sign) will match any upper or lower case letter, a `?` (question mark) will match any character and `%` (percent sign) will match any character that is not a letter or digit. E.g. a trigger text of `System Error (@####)` will match the text "System Error" followed by a space and an error code consisting of a letter and four digit number in brackets.
Note: Each placeholder will match exactly one character (i.e. there is no equivalent to the `*` placeholder as it exists in filenames or regexp). Further, there is a restriction that the last character of a trigger text may not be a placeholder.

Regular Expressions:
Note: This function is still experimental.
The search text can also be a simple regular expression. This function is costly in terms of CPU (please limit your use of regular expressions if you expect large bursts of output, e.g. `ls -R` commands on a Linux shell).
You will need to provide a regular expression (see below), together with a boundary characters class and the maximum length of text that the regular expression is matched against.
Characters of the boundary class are expected to surround the potential result. E.g. if you select whitespace characters as the boundary class, then your regular expression will only match, if it has whitespace character on each side (a beginning or end of a line matches all these classes). For easier understanding, imagine that a single character of this class will be added to the left and right of your regular expression. E.g. a pattern of `\d+` with boundary class of whitespace will match against the number `999` in `a = 999 + b;`, but not in `a= 999+b;`.
The maximum length determines the maximum length that you expect to match. Longer pieces of text, even if they occur between the correct potential boundary characters, will not be matched. E.g. a pattern of `\d+` with boundary class of non-digits and maximum length of 8 will match against the number in `a= 123;`, but not in `a= 333666999;`.

Regular Expressions Elements:
The regexp search supports only the most basic regular expression types.

`.` Dot, matches any character
`*` Asterisk, match zero or more (greedy)
`+` Plus, match one or more (greedy)
`?` Question, match zero or one (non-greedy)

`[abc]` Character class, match if one of {'a', 'b', 'c'}
`[^abc]` Inverted class, match if NOT one of {'a', 'b', 'c'}
`[a-zA-Z]` Character ranges, match any character of the ranges a-z or A-Z

`\` exactly the character that follows next, e.g. `\*` will match a `*`-character. Exceptions are the characters `aAsSwWdD` (see below), where the combination represents whole classes of characters.

\s Whitespace and any control characters (hex 00 - 1F)
\S Non-whitespace
\w Alphanumeric/Words [a-zA-Z0-9_]
\W Non-alphanumeric
\d Digits, [0-9]
\D Non-digits
\a alphabetic characters, latin letters [a-zA-Z]
\A Non alphabetic characters

^ left anchor not supported
$ right anchor not supported
(a|b) parantheses/or-expression not supported
{m,n} bracket with counting not supported

Regular Expressions Examples:
\d+\.\d+\.\d+\.\d+ Something that roughly matches an IP address (use non-digit boundary characters).
\a\a\a\a\d\d\d\d An error code of four letters and four digits (use non-alnum boundary characters).

1.7.3.16 File Transfer

These settings define the characteristics of file transfers. They are accessed via the Options menu or by clicking the file transfer protocol button in the status line.

PROTOCOL
Select X- Y- or Zmodem protocol or any of the other protocols offered here. The selection will become the active file transfer protocol whenever this session profile is activated.

ZMODEM

ZModem is one of the most commonly used file transfer protocols. It allows the transfer of files over almost any kind of connection and handles error-correction, bulk file transfers and preservation of file attributes very well. Whenever available, the use of Zmodem should be strongly considered.

See Zmodem Transfer for a description of its options and how Zmodem is used. There is also information about how it can be installed on Linux computers.

SCP

The SCP file transfer protocol will only work with SSH connections and all you need to do to send or receive a file to the remote host, is to log on to a server and to select Upload or Download from the Transfer menu (see SCP Transfer).

KERMIT

Kermit is an ancient file transfer protocol. Benefits are wide cross-platform availability and good reliability over bad connections.

Normally the standard option values should work fine (press the Reset button to set these). The block size can be any number from 80 to 9024. If you are working in an environment that allows only 7bit transmissions, uncheck the 8bit option.

IND$FILE

The IND$FILE file transfer protocol is for use in conjunction with TN3270 host connections to IBM mainframes (see IND$FILE Transfer for details).

YMODEM

Some systems offer protocols named Ymodem and Ymodem batch, which really are Xmodem with

Block-1024 (incorrectly named Ymodem) and Ymodem (incorrectly named Ymodem-Batch). In those cases, set ZOC to Xmodem with 1KB-Blocks for Ymodem and/or set ZOC to Ymodem when the host requires Ymodem-Batch.

XMODEM

These options only apply when the Xmodem protocol is selected.

CRC on

CRC is controlled by the receiver of a file. CRC can be used instead of the slightly less reliable Xmodem checksums.

Blocksize 1024

With this option Xmodem uses blocks of 1024 instead of 128 bytes. Some systems falsely call this Ymodem. The block size is controlled by the sender.

Chop pads

If enabled, ZOC tries to remove padding bytes at the end when receiving a file.

TRANSFER BEHAVIOR

If possible, detect and start file transfers automatically

If enabled, ZOC will try to automatically detect when the remote host initiates a file transfer. This is done by checking the incoming data for sequences that are unique to some file transfer methods (e.g. Zmodem or Kermit), but will not work with all file transfer methods.

Note: This option should be disabled if you intend to perform file transfers through the REXX scripting language.

Close file transfer window immediately after completion

After a file transfer completed successfully, this option turns off the 3 seconds waiting time before closing the transfer window.

Do not write transfer summary to terminal screen

If this option is set, the program will not print the transfer summary, which is normally displayed in the terminal are of the ZOC window (alongside the output of the remote host).

See also: Session Profiles, Customizing ZOC and Options Menu

1.7.3.17 File Handling

These settings define the characteristics of file handling.

IF FILE EXISTS ...

Select what you want to do if you try to download a file that already exists in your computer's download directory. You can choose different actions in case the incoming file has a modification date that is older, same or newer than the existing one, or if the incoming file has no modification date at all (e.g. when transferred with Xmodem).

Additionally you can specify, if you want the old (existing) or the new (incoming file) renamed (if rename is chosen somewhere).

SPECIAL FILE EXTENSIONS

ZOC lets you list extensions of files that should be downloaded to an alternate directory (see Options→Program Settings→Folders) and files that should be deleted after they were uploaded successfully.

The file extensions may contain the * and ? wildcard characters and need to be separated by vertical bars, but without space characters, like this: `GIF|Q*|MO?|TU?`

Note: Specify file extensions only, not full file names (i.e. '*.GIF|ABC.DOC' will not work)

Note: If you want to implement a more complex scheme to manage downloaded files, you can use the ZOC-Events REXX file (see Options→Program Settings→Special Files).

MISCELLANEOUS

Show JPG/GIF pictures while downloading

ZOC is able to show pictures while downloading them. For this purpose the file SHOWEM.DLL are provided in your ZOC directory. If you do not like this feature, it can be disabled.

See also: Session Profiles, Customizing ZOC and Options Menu

1.7.3.18 Idle Action

IDLE ACTION

Here you can set an action to be executed when no characters have been sent or received for a specific period.

If you want to prevent a disconnection by the remote site due to inactivity, it is possible to automatically send characters or perform another user-defined action in this case (e.g., start a script or send a special key of an emulation).

When sending characters, you should first try to use characters that do not trigger unwanted actions on the remote site, such as the null character (hex 00), Esc, or if necessary, Enter. For Telnet connections, the TELNET_NOP packet can also be used for this purpose.

Alternatively, ZOC can disconnect the connection after inactivity so that, for example, modem connections are automatically limited in duration.

To perform no action on a timeout, the timeout period must be set to zero or you need to set the custom action to 'no action'.

Note: SSH and Telnet connections have additional options (keep alive) which are specific to those connection type. You will find those in the options dialogs for the SSH and Telnet connection types. See also Tutorial: Configure Connection Type Options.

1.7.3.19 Trace/Debug

Here you can enable various debug trace functions. These are mainly used by technicians and developers of applications and devices, and they are mostly used to debug a connection or to provide a way to see

the communication with a remote host/device in raw form.

Show control characters in terminal
Show received control-codes (hex 00 to 1F) as plain text (e.g. `^I` for Tab) instead of performing the functions which are usually associated with these codes.

Show incoming data as hex dump in terminal
This option shows incoming characters as combination of hexadecimal values and characters.

Show data as hex dump in data stream viewer
This option shows incoming characters as hex dump in the data stream viewer instead of the terminal window.

Show data as ascii-trace in data stream viewer
This option shows incoming characters as an ascii trace in the data stream viewer instead of the terminal window.

Write ascii trace file
Creates a trace file named `zoctrace.txt` in the logging folder, which will then store all sent and received characters combined with time stamps. The trace file is formatted in human readable form (mixed text, control code names, hex codes).

You can access the logfiles by choosing Show Logfiles Folder from the Logging menu.

Write binary trace files
Creates a pair of trace files in raw binary form, containing all sent characters (`zoctrcout.bin`) and all received characters (`zoctrcin.bin`).

You can access the logfiles by choosing Show Logfiles Folder from the Logging menu.

Write low-level device trace files
This option also creates a pair of trace files in raw binary form. Depending on the communication method, these files will also contain control characters that are used by the communication layer internally. E.g. for telnet connections, the low-level trace file will also contain the telnet protocol negotiation elements (e.g. IAC sequenced and end-of-record markers).

Write debug-log for EmTec-support
This protocol is of little use unless you will be asked by EmTec support to use this option.

See also: Session Profiles, Customizing ZOC and Options Menu

1.7.3.20 Advanced

Activate control sequences that enable remote hosts to execute commands locally
This option enables those of the extended control sequences, which are normally disabled because they allow execution of commands on the local computer and hence pose a security risk. It goes without saying that this option, should only be enabled if you *absolutely* trust the host to which you are connected.

When files are dropped over the terminal
If files are dropped over the terminal window, you can choose to either transfer this files (or the first file if the current file transfer does not support multiple file transfer), or to send the filenames through the terminal.

Note: emulations (TN3270, TN5250).

1.7.3.21 Handling Fonts, Window Size, Colors

The text handling of a terminal emulation differs noticeably from a text processor. The topics below explain the way ZOC deals with the issues regarding fonts, window size and color.

Fonts

Terminal emulations can only use fonts where all characters have the same width and height, i.e. where the letter `i` has the width as the letter `W`. This is necessary to display tables or indentations correctly. Under Windows this is true for the Lucida Console or Courier New fonts.

Character Set

Additionally the host expects that certain characters are displayed when a certain code is sent to the terminal. For example some hosts send a code to display line graphic characters while others may send the same code to display a European language character.

The mapping of characters to codes is defined in so called character sets. Most hosts either expect the Windows/Ansi/Latin or the IBM/DOS character set.

However, this setting does not apply to all terminal emulations. For example, the IBM terminals are based on the EBCDIC code, therefore they have their own NCRs (National Character Sets) which differ from most other emulations (which are ASCII based).

Hence the TN3270 and TN5250 emulations, these ignore the character set setting here and instead have a national character set, which is part of the specific settings for these emulations.

The VT220 emulation also has specific NCRs and the remote host can even switch/change them throughout a session.

Fonts and character sets are configured in Options, Session Profile, Layout

Window Size

Most remote hosts display text in a fixed number of lines with a fixed number of characters per line. Usually 25 lines of 80 characters are used (this may differ depending on the host and emulation). Together with the fact, that all characters in a font have the same width and height, this determines the screen space necessary to display the terminal area. 25 lines of 80 characters with a small font will take up less space than 25 lines of 80 characters with a larger font. ZOC will adjust its window around the space which is necessary to display all lines and characters with a given font. This means that the size of the ZOC window depends on these factors.

The ways how font selection interacts with window sizing can be configured in Options, Session Profile, Layout and partly relies on the terminal size configuration in Session Profile, Terminal Characteristics

Colors

Most emulations work with a set of 16 base color (black, red, blue, green, ...). In Options, Session Profile, Window you can select the colors, which ZOC will use for text and background. However, the text and background colors can be changed by the remote host (e.g. it may switch to yellow to display highlighted text). Remote hosts usually assume gray text on black background so the colors which the remote host selects, may or may not be good to read on your individual configuration.

In the window configuration it is also possible to change the 16 base colors. When this is done, ZOC will display your selected color when the remote application requests one of the 16 base colors. Usually the global color table is only changed to modify the color tones of a given color, e.g. to make the standard green a bit brighter or dimmer or to give it a different tone of green. However, taking this to the extreme, it is also possible to select a completely different color, so that ZOC displays orange when the remote host requests green. Naturally this may lead to surprising results.

In the TN3270 and 5250 emulations, the text and background color settings are ignored. The reason for this is that these emulations work based on functional colors, i.e. the host will request highlight color, status color, message color, etc. from the terminal. Due to the vast number of combinations possible, ZOC comes with a selection of color schemes in which colors have been selected for good readability. You can select one of these color schemes for the 3270 and 5250 emulations in Session Profile, Emulation. The color schemes are built from the 16 base colors, so it is possible to change them by changing the global color table in Session Profile, Window

See also: Session Profiles, Customizing ZOC and Options Menu

1.8 Connection Types (Input/Output Methods)

ZOC supports a variety of communication methods through which you can communicate with remote hosts. Here is a list of these methods descriptions of their details, e.g. options, usage, etc.

- Secure Shell
- Telnet
- Telnet/SSL
- Serial/Direct and Serial/Modem
- Windows-Modems
- Local Shell (Windows Command Prompt)
- Named Pipe
- Rlogin
- UDP (Datagram)

1.8.1 Secure Shell (SSH)

Secure Shell is a communication mechanism that uses an encrypted connection over unsecured networks.
The following topics cover various aspects of using SSH in ZOC:

- Overview
- Options for Secure Shell connections
- SSH Global Authentication Keys
- SSH Menu Functions
- SSH File Transfer
- REXX ZocDeviceControl command for SSH
- Create SSH Public/Private Key Files
- Changing Passphrases
- Manage SSH Tunnel Profiles

1.8.1.1 SSH (Secure Shell) Overview

Introduction
The secure shell communication allows you to login to a host via internet/LAN. Unlike using Telnet, an SSH connection uses encryption for all data sent and received, which makes it impossible for third parties to tamper with or monitor the connection to obtain passwords or confidential data.

Note: ZOC's implementation of the Secure Shell protocol is based on the ssh client made by the OpenSSH project. You will find more information about OpenSSH on http://www.openssh.org

Login
To authenticate with the remote host, you can use the public/private key, username/password and/or keyboard-interactive methods. If you want to login via public/private key, you need to provide the names of the key files for your host before making the connection (enter them in the host directory entry or in the quick connect dialog or in SSH's global options). For username/password or similar methods you can either provide these upfront or you will be asked to provide them when necessary.

Note: Please also read SSH Key Generator for information about how to transfer the public key file to the server.

File Transfer
To transfer files over SSH you can either use SCP or Zmodem. See SSH File Transfer for more information.

1.8.1.2 Options for SSH Connections

To adjust options for SSH connections, navigate to the Connection Type section of the Session Profile. These changes will apply to all connections using that session profile.
If you want to change the options for a single connection, click the Configure button in the Quick Connect window. For connections made through the Host Directory, use the the Configure button found in the Host tab of your host directory entry.

The following options are available when configuring Secure Shell sessions:

Connect via Proxy
If you need to connect to your SSH host through a proxy (typical proxies are HTTP, SOCKS), you can enter the type, the name/ip and optional port (default port is 1080) in the proxy field of the SSH options dialog, e.g. `192.168.1.1` or `myproxy.somewhere.com:8080`

If the proxy is a SOCKS5 or HTTP type, you can also provide username and password in the form user:pass@host:port.

A jump server (sometimes also called bastion host)is a ssh host, that is used as a gateway to the final destination. Technically this involves an initial ssh connections to the jump server. After being logged in there, a ssh command to connect to the final server is executed on the jump server. If the jump server requires a different username than the file host, you can provide it in the form user@host:port

Check host key against known hosts file
As an additional security feature, ZOC's SSH implementation offers a function to check the remote host's encryption key with that of earlier sessions with the same host. The list of known keys is stored in the files named known_hosts (for SSH V1) and known_hosts2 (for SSH V2) in ZOC's SSH directory. These files are compatible with the respective files of various Unix implementations of SSH.

Send keep-alive signal to server
Enabling the 'keep alive' option is intended to prevent the server from terminating the connection due to inactivity. When the option is set, ZOC will configure the TCP connection with SO_KEEPALIVE and will also send application level SSH keepalive packets (this is equivalent to setting both keepalives and ServerAliveInterval=60 in OpenSSH).

Show password prompts in the terminal
When the server requires passwords or passphrases ZOC normally shows a pop-up dialog to enter those. With this option, the prompts are instead shown in the terminal area in the same way as it happens in the OpenSSH `ssh` command.

Edit Global Authentication Files
As an alternative to the username/password authentication, it is possible to use global authentication

files. Authentication files contain public and private keys and can be used to authenticate you when logging on to a host. If you specify global authentication files, these will be used for all SSH connections. This is useful if you use the same authentication file for all your hosts. Additionally you will be able to specify authentication files on a per connection basis (this can be done in the Quick Connect dialog or in the host directory). See also: SSH Communication→Global Authentication Files

ADVANCED OPTIONS

Authentication Methods

The available authentication methods can be enabled or disabled. ZOC Terminal and the SSH server will then try to find common methods and will see if the user can authenticate through one of them. The order in which the methods are tried (if enabled) is: gssapi, publickey, keyboard-interactive, password. If one of the options is set to "Preferred", it will be moved to the front of the list.

Public-Key Authentication

Enables or disables the public-key authentication method. This method is based on public-private key file pairs to prove your identity.

Agent Support

If one of these options is active, ZOC will contact an internal or outside agent to provide private keys for login or for ssh connections which are initiated from within the original session (the latter is called agent-forwarding).

When the choice `internal` is enabled, the private-key which is used to log into the host (if private-key authentication is use at all) will be offered to also authenticate further `ssh` commands that are issued within the original session. This is (in a limited way) similar to what OpenSSH's `-A` option does in combination with ssh-agent, but doesn't require keys to be loaded into an external agent.

The other options let you choose, if either Putty-Agent (pageant.exe), or the ssh-agent (Windows OpenSSH or macOS), or the zoc-agent (ZOC' Tools menu) should be contacted to provide private keys for login and/or for inner ssh sessions (as in OpenSSH's `-A` agent-forwarding option).

GSSAPI Authentication

Enables or disables the GSSAPI authentication method. This method is usually used for an authentication type, where the ssh server contacts a central host to determine if the user has permission to log in.

GSSAPI Authentication Types

When GSSAPI authentication is active, you can choose between two implementations: Kerberos GSSAPI (using the GSSAPI library from the original MIT Kerberos packet) or Microsoft-SSPI (which refers to the Microsoft implementation of the Kerberos protocol, which usually goes along with a Windows Domain account and Active-Directory).

Keyboard-Interactive Authentication

Enables or disables the keyboard-interactive authentication method. This method is normally used for challenge-response authentication (e.g. with SecureID cards). Sometimes it simply prompts for the password.

Password Authentication

Enables or disables the password authentication method.

Tunnel profiles

If you need port forwarding (tunneling) you can use the push button to define so called tunnel

profiles. These profiles allow you to build combinations of local and remote port numbers which will be forwarded to a host on the other side of the connection. You will then be able to select such a profile when making a SSH connection (see also Manage Tunnel Profiles).

Enable X11 connection forwarding
Creates a tunnel that will forward X11 data from server to client. This option is required if you want to run X11 commands on the remote shell. Equivalent of OpenSSH's `-X` parameter.

Remote Command
A command that will be executed remotely instead of connecting to a shell. This is equivalent to the 'command' parameter in the OpenSSH ssh command.

Other Options
Since ZOC's SSH implementation is based on OpenSSH, command line parameters from OpenSSH could be added here, e.g. `-4` to limit connectivity to IpV4 or `-C` to use gzip compression on the data channel. Please be aware that not all options are supported, especially if they are related to more advanced features like configuration files, proxy commands, multiplex masters, etc.

1.8.1.3 Global SSH Authentication Files

Global authentication files are useful if you connect to a variety of hosts, where you use the same public/private (identity) key files for authentication.
These files are used for authentication, when you make a connection via Quick Connection or via Host Directory and if you choose 'Global SSH Key Files' there for the ssh-key field.
Alternately, you can set the option to always use the global key files for all connections, so that you will not have to choose this option for each individual connection.
The Global Authentication Files dialog is available through the Tools menu.

See also: SSH Communication→Options

1.8.1.4 SSH Menu Functions

When active, the SSH connection type adds three functions to ZOC's File menu:

- Create SSH Public/Private Key Files
- Changing Passphrases
- Manage SSH Tunnel Profiles).

1.8.1.5 SSH File Transfers

The best ways to transfer files over an SSH connection are SCP or Zmodem. For both methods you will need to switch ZOC to the corresponding file transfer protocol: Either go to Options→Session Profile→Transfers and activate SCP or Zmodem. Alternately, if you are connecting to the host from an entry in the ZOC host directory, you can select the file transfer protocol in the entry's Options tab.

SCP
Connect to a remote server via SSH and login to the shell prompt. To send a file (or multiple files) from your computer to the remote host, select the Upload function from ZOC's Transfer menu. Alternately, to transfer a file or folder from the remote computer to your local computer, select the Download function. The source/destination folder on the remote computer will be your current working directory there. For the local computer, the default up/download folders are configured in Options→Program Options→Folders.
See also: SCP Transfer
Zmodem

Most Unix/Linux systems haven an implementation of Zmodem installed and preconfigured (if not, look for a package named lrzsz.zip on the internet or ask our support). ZOC and rz/sz works very well over a SSH connection and integrates seamlessly with ZOC.
See Zmodem Transfer for a description of how Zmodem is used from a Linux command line and for a description of options.

1.8.1.6 REXX ZocDeviceControl Command for SSH

`ZocDeviceControl` is a ZOC-REXX Command, which allows to perform functions that are specific to a communication method.
Available sub-commands for the `ZocDeviceControl` script command in relation to SSH are:

RESOLV <hostname or ip>
Returns hostname and ip for the requested host in the form "<primary hostname>; <ip>" (e.g. `SAY ZocDeviceControl("RESOLV mail.emtec.com")` might return `www2.emtec.com; 212.34.172.147`).

1.8.1.7 Create SSH Public/Private Key Files

Note: This window offers a convenient way to create the most commonly used public/private keys (including non-resident FIDO2 keys). If you need more fine grained control over the key creation, you will find a command line program in the ZOC executable folder which mimicks the exact behavior of the OpenSSH ssh-keygen command, including support for the ecdsa-sk and ed25519-sk key types. This program is located at `C:\Program Files\ZOC9ssh-keygen.exe`

The 'SSH Create Public/Private Key Pair' window lets you create authentication files to identify you with hosts that offer public/private key file authentication.
You need to specify the type of public/private key to create and the length of the key in bits (usually 2048 or more) and the key type (the latter depends on your host and the SSH version which the host offers).
The passphrase is a word or a short sentence which is used to protect the key in order do avoid that someone who gets hold of your private key file can use the key to log on with your accounts.
After the keys are created, they are stored in two files, where the public part of the key ends with the extension `.pub`.
The private key will need to remain in ZOC's SSH directory and should not be shared with others.
The public part needs to be transferred to your SSH server were it is usually appended to `authorized_keys` file in the `~/.ssh` directory.

With ZOC this can be done on a Unix/Linux host using the following way:
1. Copy the contents of the newly created public key file, e.g. `id_rsa.pub` file to the clipboard by using the Function Tools-menu→Copy SSH Public-Key to Clipboard.
2. Log on to the remote computer and type the following commds, where after typing `echo` you paste the contents of clipboard into the command by using the Edit-menu→Paste function:

```
cd ~/.ssh
echo (paste clipboard here) >>authorized_keys
```

See also: SSH Communication→Options and SSH Communication→Global Authentication Keys.

1.8.1.8 Changing Passphrases

The private part of your authentication files will be protected by a passphrase (usually a short sentence which is used like a password). If necessary, this function allows you to change the passphrases for your files.
See also: SSH Communication and Creating Keys

1.8.1.9 Manage SSH Tunnel Profiles

Tunnel profiles are used to route data from a local port to a remote host or remote port to a local host through the secure SSH connection. A profile consists of one or more specifications of port/host pairs and can be assigned to a connection.
This dialog lets you manage tunnel profile files. You can load an existing file, create, edit or delete entries, save it or save it under a different name. To use the tunnels in the profile, the name of the file can then be specified in the SSH options of a connection.
If, for example, you wanted to do secure mail checking, you could install a tunnel from your local port `10110` to your mail server `mail.hogwarts.edu.110` (see Tunnel Profile Items).
In your email program you would then configure the account to connect to `localhost:10110` or `127.0.0.1:10110` (which both refer to your own computer). Assuming there is a connection to a host active with the given tunnel profile, ZOC would then route the connection through the SSH stream to the remote mail server.
Additionally ZOC also supports dynamic port forwarding, which essentially turns the SSH connection and ZOC into a SOCKS4 or SOCKS5 proxy.
See also: SSH Communication→Options and Tunnel Profile Items

1.8.1.10 Tunnel Profile Items

As described in Manage Tunnel Profiles, a tunnel consists of a port number on one side and a combination of computer address and port on the other.
This dialog lets you enter the data for one tunnel, e.g. the data from the example above would be:
Tunnel from local to remote address.
Local port: `10110`
Remote address: `mail.hogwarts.edu:110`
Additionally ZOC supports dynamic port forwarding, which essentially turns the SSH connection and ZOC into a SOCKS4 or SOCKS5 proxy. In these cases there is no destination address and port, because the calling program will define those on the fly when making the connection.

Note: As with OpenSSH, IPv6 addresses need to be enclosed in square brackets, e.g. `[fc00::100:1]:10022`

1.8.1.11 Selecting a SSH Key

When selecting an SSH key, you can choose between various options.

Key File(s)
This will accept OpenSSH compatible private key files or certs. If you have a Putty ppk file, you will need to use putty-keygen to convert it into OpenSSH format.

PKCS#11 Library
PKCS#11 is an interface mostly used with card readers or other hardware keys. You select a Windows DLL or macOS dylib/so file which allows the ZOC Terminal to access the hardware keys. For example, for the relatively popular Yubi-Key plugs, the corresponding file would be `C:\Program Files\Yubico\Yubico PIV Tool\bin\libykcs11.dll`.

Global SSH Keys
ZOC allows to define global authentication keys via its Tools menu or through the SSH options window. When you choose this option, ZOC will use all of the keys defined there and will offer them to the server for authentication.

Windows Certificate
If you have authentication keys in the Windows Certificate Store, this option lets you choose one. This also applies to hardware, e.g. CAC/PIV cards or usb hardware keys which map their keys to the Windows Certificate Store.

1.8.2 Telnet

The Telnet connection type allows access to remote hosts through an internet or LAN connection using the so called Telnet protocol.
Telnet transmits unencrypted data and has been mostly replaced by the Secure Shell protocol.
The following topics cover the use of telnet connection in ZOC:

- Telnet Overview
- Options for Telnet Sessions
- Menu Functions
- REXX ZocDeviceControl command for Telnet

1.8.2.1 Telnet Overview

The Telnet connection type allows access to remote hosts through an internet or LAN connection using the telnet protocol.
To make a connection to a host you provide the host name or internet address in the Connect To field, e.g. `bbs.hogwarts.edu`. ZOC will then try to connect to the host using your internet connection.
You will be connected to the standard Telnet port (23) of the remote host. If you want to connect to different port, you may provide the port number or service name after the host name or address separated by a colon or space. Examples: `mail.hogwarts.edu 25` or `mail.hogwarts.edu:25` or `mail.hogwarts.edu smtp`.
If the remote host is accepting TCP stream connections, but if it is not running the full telnet protocol (e.g. SMTP or other RFC protocols) you can enable the Pure Socket Connection option in the Telnet settings.
In some cases, when you get an `NO RESPONSE` error you can try to put an * (asterisk) in front of the host name (or address). This way ZOC will not try to ping the host before making the call (this will help for hosts which take very long to respond): `*slowhost.leisure.net`

1.8.2.2 Telnet Options

To adjust options for telnet connections, navigate to the Connection Type section of the Session Profile. These changes will apply to all connections using that session profile.
If you want to change the options for a single connection, click the Configure button in the Quick Connect window. For connections made through the Host Directory, use the the Configure button found in the Host tab of your host directory entry.

The following options are available for telnet connections:

Socks Proxy
If you need to connect to your telnet host through a proxy you can enter the name/ip and optional port (default port is 1080) here. If the proxy is a SOCKS5 type, you can also provide username and password in the form user:pass@ip:port.

Raw socket connection, pure TCP, no telnet
This option generates a pure TCP connection without negotiating telnet options with the remote host. This is useful if the server is not a real telnet server but runs a different TCP based protocol instead (e.g. SMTP or other RFC protocols).

Enable TCP Keep Alive

When this option is set, the telnet tcp stream is configured to send keep-alive packets to the remote hosts (technically this means that SO_KEEPALIVE is set on the socket). In addition to that, a TELNET NOP packet (FFh F1h) is sent every 60 seconds.

If this does not work, you can also set the idle feature in the Terminal options of the Session Profile (see this article Tutorial: Configure Timeout Settings).

Start session with local echo on
If this option is enabled, ZOC will use local echo (which is faster than remote echo) when the session starts. However, during the login process this option is often overridden (i.e. changed back to remote echo) by the remote host.

Host may change terminal's local echo mode
Allows the host to change your local echo setting (Session Profile→Terminal).

Do not disconnect on error
If a transmission error occurs, ZOC normally closes the connection. With this option checked, ZOC will keep the connection alive.

ADVANCED OPTIONS

Do not lookup hostnames from IP address
If you connect to a host by using its IP ZOC normally looks up the host name. You can turn this off, if you do not want this (e.g. because your name server is slow or if you are connecting to ip addresses which do not have names associated with them).

Connect IP-V4 only
If this option is enabled, DNS lookup will use IP-V4 addresses only.

Outgoing CR conversion
Enable this option if your host requires a NUL character sent after a CR character if the CR character isn't followed by a LF character.

Host sends CR/NUL in binary mode
Try to enable this option if binary file transfers (e.g. Zmodem) through Telnet connections fail.

Offer 'do SGA' negotiation
Please enable this option if you get errors from the host saying that the client does not support SGA.

Start session in binary mode
If enabled, ZOC will try to negotiate a binary session. This is sometimes necessary for file transfers to work.

Telnet Accept will listen on port
If you use the Accept Connections function from the File menu, this is the number of the port on which ZOC will listen for incoming connections.

Terminal Speed
Here you can enter the terminal speed in baud, if needed (e.g. 9600,9600).

1.8.2.3 Telnet Menu Functions

The Telnet handler adds three functions to ZOC's File menu.

Are you there?
This function tries to trigger a response from the remote host (usually something like `[YES]`) to find out you are still connected to the host.

Interrupt Process
This function aborts the program that currently runs on the remote host. It is similar to pressing Ctrl+C or Ctrl+Break locally.

Abort Output
This function aborts output from the remote host (e.g. when accidentally showing a large text file).

1.8.2.4 REXX ZocDeviceControl Command for Telnet

`ZocDeviceControl` is a ZOC-REXX Command, which allows to perform functions that are specific to a commnunication method.
Available sub-commands for the `ZocDeviceControl` script command in relation to Telnet connections are:

RESOLV <hostname or ip>
Returns hostname and ip for the requested host in the form "<primary hostname>; <ip>" (e.g. `SAY ZocDeviceControl("RESOLV mail.emtec.com")` may return `www2.emtec.com; 212.34.172.147`).

TCP_NODELAY ON
Enables the TCP_NODELAY option on socket level (google for TCP_NODELAY or Nagle's algorithm), e.g. `CALL ZocDeviceControl "TCP_NODELAY ON"`

1.8.3 Telnet/SSL

Telnet/SSL is a communication mechanism that uses an encrypted Telnet connection over unsecured networks. The protocol is basically Telnet, but wrapped into an SSL encryption layer.
The following topics cover various aspects of using Telnet/SSL in ZOC:

- Telnet/SSL Overview
- Options for Telnet/SSL Sessions
- REXX ZocDeviceControl

1.8.3.1 Telnet/SSL Overview

The Telnet/SSL connection type is basically identical to the Telnet communication method, with the exception that all traffic runs through an TLS/SSL encryption mechanism.
The method is useful if you want to connect to SSL based hosts (e.g. TLS/SSL secured IBM Mainframes or web servers on port 443).

1.8.3.2 Telnet/SSL Options

To adjust options for telnet connections, navigate to the Connection Type section of the Session Profile. These changes will apply to all connections using that session profile.
If you want to change the options for a single connection, click the Configure button in the Quick Connect

window. For connections made through the Host Directory, use the the Configure button found in the Host tab of your host directory entry.

Most options are the same as for Telnet (see Telnet Options).
The Telnet/SSL handler is basically identical to the Telnet handler with the exception that all traffic runs through an SSL encryption layer.
Telnet/SSL supports client certificate authentication. If the certificate file does not contain a private key, a separate private keyfile can be entered.
Also you can force the security protocol to TLS 1.2 or TLS 1.3. Otherwise ZOC will also accept SSLv2, SSLv3, TLS 1.0 and TLS 1.1 as possible security protocols.

1.8.3.3 Telnet/SSL ZocDeviceControl

`ZocDeviceControl` is a ZOC-REXX Command, which allows to perform functions that are specific to a commnunication method.
Available sub-commands for the `ZocDeviceControl` script command in relation to Telnet/SSL are the same as for Telnet.

1.8.4 Serial/Direct and Serial/Modem

Serial/Direct and Serial/Modem are communication mechanisms within ZOC which use local serial connections (Serial/Direct) or long distance serial connections (Serial/Modem) to communicate with equipment or hosts.
The following topics cover aspects in relation to using serial connections within ZOC:

- Serial Overview
- Options for Serial Connections
- REXX ZocDeviceControl
- AT Commands

1.8.4.1 Serial/Modem and Serial/Direct Overview

This section applies to two similar connection types: Serial/Modem and Serial/Direct
The purpose of Serial/Direct is to connect to gear which is hooked directly to the serial port of your machine, e.g. routers, flash programming devices and such.
The Serial/Modem method should be used to dial out to a remote location using a modem which is attached to the serial port and where the modem is controlled through means of AT-commands (e.g. ATDT <number> to dial).

The two are similar in that they work through serial ports and have almost the same options, but there are two basic differences between Serial/Modem and Serial/Direct:
1) Serial/Modem will internally issue AT commands to the attached port in order to configure the modem it and to let it make the connection. Serial/Direct does not send any commands or text strings by itself.
2) Serial/Modem will operate under the assumption that 'connection' refers to a connection to a remote location which the modem has made (not connecting to the serial port talking to the modem itself). With Serial/Direct however, 'connection' means opening the serial port in order to exchange data with the attached device itself.
If you have preconfigured modems (or modem-like devices like cellular phones) attached to your computer, it will probably be easier to use them in ZOC as Windows-Modems rather than using Serial/Modem, especially if you want to use these to make connections to remote computers (e.g. using a

cellular phone to dial into an office computer) rather than sending commands to them directly (e.g. in order to access special features of the cellular phone itself).
For details about how to connect to a remote host via ZOC's Serial/Modem and/or how to use Serial/Direct to talk to devices which are attached directly to a serial port, please follow the instructions in Quick Start Guides

1.8.4.2 Serial/Modem and Serial/Direct Options

To adjust options for serial connections, navigate to the Connection Type section of the Session Profile. These changes will apply to all connections using that session profile.
If you want to change the options for a single connection, click the Configure button in the Quick Connect window. For connections made through the Host Directory, use the the Configure button found in the Host tab of your host directory entry.

The following options are available when configuring serial connections:

Com-Port

Enter the name of the communications port in this entry field. This is usually a string like `COM1` under Windows (note that there is no space between COM and 1) or `/dev/cu.pl2303serial` under macOS. To access com ports above COM8 on Windows NT or higher use \\.\COMxx. To use remote COM-ports in a Windows Network, a modem sharing tool (e.g. Stomper) is required.

Serial Options

In this section you define how data is transferred between your computer and the modem (which is not necessarily the same way the modem uses to transfer data across the telephone line). You have to select a speed (bits per second), the number of data bits, a parity mode (none, even, odd, mark or space) and the number of stop bits.

What you use depends mostly on your host.

These settings are often given (e.g. in instructions of how to call a host) in an abbreviated version, like 38400-8N1 (38400 bits per second, 8 data bits, no parity, one stop bit).

RTS/CTS handshake

If enabled, ZOC uses RTS/CTS hardware handshake for communication with your modem.

RTS/CTS is used to control the flow of data between the computer and the modem and provides a way for both to prevent the other from sending data.

This is essential for file transfers when the speed of the modem to modem connection is different from the modem to computer connection (which is true for MNP5 of V.42 modems). It is also essential in multitasking environments where the processor might have other things to do when data arrives.

Thus, it is highly recommended to have this option enabled. However, the modem or attached device needs to support it as well, so check your modem manual for the proper modem command to "enable bi-directional RTS/CTS hardware flow control" for use in the modem init string (see Modem Options).

DSR handshake

This DSR/DTR handshake is somewhat similar to RTS/CTS. However, it does not control the data flow, but the general availability of the modem and computer (i.e. if these are turned on).

If enabled, ZOC monitors the DSR signal for communication with the modem. This option should only be enabled if your modem and your cable properly provide the DSR signal.

XON/XOFF

If enabled, ZOC uses the Xon/Xoff software handshake for communication (which is yet another

method to control the data stream). It uses special characters to hold/release the sender. This method is inferior to RTS/CTS handshaking and should only be used if necessary.

Valid CD signal

If your modem supports the Carrier Detect (CD) signal (most modems will do this if you add AT&C1 to the modem init string in the Modem Options), you should enable this option (which is highly recommended).

ZOC uses the CD signal to control the connection timer and other functions which depend on if a connection is active or not.

Break signal duration

The duration that is used for sending a break signal (some hosts use this to break an operation) when pressing Ctrl+End. It is given in milliseconds and typically ranges from 250 to 400 ms.

AT-Commands

The Serial/Modem method lets you create and select multiple profiles with AT commands to control modem functions like dial, hangup etc. You can select an existing profile, change the AT commands or leave the field empty to use ZOC's defaults.

Release port if window is minimized

Additionally you can set an option to release the COM port when the ZOC window is minimized. This would allow you to access the COM port from another software while the ZOC windows is minimized. Please note "CD Signal is Valid" option must be activated and the port will only be released if there is no active modem connection.

1.8.4.3 REXX ZocDeviceControl

`ZocDeviceControl` is a ZOC-REXX Command, which allows to perform functions that are specific to a commnunication method.
Available sub-commands for the `ZocDeviceControl` script command in relation to serial connections are:

GETRS232SIGNALS

Return a string that indicates the raised incoming signals (CTS, DSR, CD, RI) on the serial port, e.g. `##[CTS][DSR][CD]##`
Note: For backward compatibility the command GETSTATE returns the same result as GETRS232SIGNALS).

SETRTS ON|OFF

Changes the status of the RTS Signals on the RS323 port (this will only work, if the Rts/Cts is not configured as 'Handshake' in the serial settings), e.g. `Call ZocDeviceControl("SETRTS ON")`

SETDTR ON|OFF

Changes the status of the DTR Signals on the RS323 port (this will only work, if the DTR/DSR is not configured as 'Handshake' in the serial settings), e.g. `Call ZocDeviceControl("SETDTR OFF")`

TESTACCESS <name>

This command tests if the serial port <name> (e.g. `SAY ZocDeviceControl("TESTACCESS COM3")` or `SAY ZocDeviceControl("TESTACCESS /dev/cu.pl2303")`) is valid and accessible.

FINDUSBPORT <text>

This command searches for the first USB port, that has <text> in the device's verbose description (e.g. `port= ZocDeviceControl("FINDUSBPORT FTDI")`)=.

1.8.4.4 Modem AT Commands for Serial/Modem

This window is used to define the strings ZOC uses to communicate with your modem. These strings may contain control characters (e.g. ^M, see Special Codes), the tilde (~, which is a delay of 1/3 sec) and ^# to be substituted with a phone number.

Initialization

This is a list of modem commands to set the modem to a defined state before using it. You should be aware that ZOC (unlike some other programs) requires a ^M at the end of the initialization string.

If you do not know what to use for the initialization string, please try either `ATZ`, `AT&F` or `AT&F1` or check the modem manual for advice

Depending on your setup, you might want to send the initialization string every time you load an options file (e.g. when having different options files for different modem configurations).

Normally ZOC does not send an init sequence if it finds a carrier detect signal from the modem (to prevent a modem reset while you are online). However, some modems provide a CD signal after power on. In this case you should enable this option (and add AT&C1 to the modem init string to tell the modem to provide a real carrier detect signal).

Dial Commands

ZOC lets you set up four dial commands for calling different types of numbers (i.e. using a credit card number to make long distance calls). You can select the dial command to be used from the Host Directory and when dialing manually (from the File Menu).

To dial via modem a comm program has to send a dial command and the phone number to the modem and has to complete this command by sending ^M. The dial commands are ATDT for tone dialing, ATDP for pulse dialing and ATDI for Zyxel ISDN modems. Hence, for the dialing fields you use the appropriate dial command, append the string ^# (which is replaced with the telephone number) and ^M

A standard dial command would look like `ATDT^#^M` (meaning that ATDT, then the phone number and then enter is sent). If you wanted to set the modem to ignore the dial tone and send a zero before the telephone number to get an outside line, you can use `ATX3DT 0,^#^M` instead (X3 will prevent the modem from waiting for a dial tone and 0, will send a zero and make a pause before dialing the phone number).

If you want to issue a modem command before dialing you should add some tilde characters between that command and the dial command (as in `ATZ^M~~~ATDT ^#^M`) to give the modem time to process the former command before continuing to process the latter.

Hangup

ZOC supports two methods of telling a modem to hang up. Using the DTR signal of the com port is the superior of the two. If you disable it, ZOC will use the ATH command.

Auto answer

Here you supply two modem commands to enable and disable the modem's auto answer mode. These are `ATS0=1^M` and `ATS0=0^M` for basically all modems.

1.8.5 Windows Modems

Windows Modems is a communication mechanism, that uses modems which are registered as such in the Windows operating system for communication with remote computers.
The following topics cover aspects in relation to using Windows Modems connections within ZOC:

- Windows Modems Overview
- Windows Modems Options
- REXX ZocDevControl

1.8.5.1 Windows Modems Overview

This connection type allows you to use the modems drivers that are installed in your Windows control panel (so called Telephony API or TAPI modems). ZOC's Windows Modems option dialog shows a list of all installed modems in your system from which you can select the one you want to use.
If you own an ISDN board, you will probably also find ISDN style modems in this list. If you have problems with these, you will find some helpful information in the common problems/questions section of this text.

1.8.5.2 Windows Modems Options

1.8.5.3 REXX ZocDeviceControl Command for Windows Modems

`ZocDeviceControl` is a ZOC-REXX Command, which allows to perform functions that are specific to a commnunication method.
Available sub-commands for the `ZocDeviceControl` script command in relation to Windows Modems connections are:

GETMODEMID <modemname>
Return a string that indicates the permanent modem id for a given modem (to use with the ZocSetDevParm command).

Example:

```
modem= "Microlink 33.6TS PnP"
id = ZocDeviceControl("GETMODEMID "||modem)
Call ZocSetDevParm "[7]"id"|0|1:"
```

1.8.6 Local Shell (Windows Command Prompt))

Within ZOC, Local Shell isn't actually a communication mechanism, but instead allows you to have a local shell session (e.g. bash or zsh) within a ZOC tab.
The following topics cover aspects in relation to using a Local Shell within ZOC:

- Local Shell Overview
- Settings for Local Shell Sessions

- REXX ZocDevControl

1.8.6.1 Local Shell Overview

The local shell connection type will create a session which opens a shell or runs a program local computer. This way you can have a local terminal sessions (e.g. bash or zsh) within ZOC, thus replacing the macOS terminal or command prompt window.
Since all sessions ZOC are connection based, when using Local Shell you will need to specify a destination to connect to even if you are not actually connecting to another computer.
Therefore valid Connect to destinations are either `localhost` or `127.0.0.1`. In this case the Local Shell handler will run the command which is specified in the Local Shell Settings, which by default is /usr/bin/login for macOS and C:\Windows\System32\cmd.exe for Windows.
To run a different command, you can modify the Local Shell options in the Connection Type section of the Session Profile. For a single connection you can use the Configure button of the Quick Connect window or in the Host tab of your Host Directory entry.
Alternately you can use `#!` followed by any system executable as the destination. For example, on macOS you can `#!/usr/bin/login -p harry` or `#!/usr/bin/zsh` or `#!/usr/bin/ssh secure.hogwarts.edu` in the Connect to field. Under Windows you can specify Connect to as a command like `#!C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`.

1.8.6.2 Local Shell Settings

To adjust options for local-shell connections, navigate to the Connection Type section of the Session Profile. These changes will apply to all connections using that session profile.
If you want to change the options for a single connection, click the Configure button in the Quick Connect window. For connections made through the Host Directory, use the the Configure button found in the Host tab of your host directory entry.

Local Shell sessions offers the following configuration options:

Local Shell

When activating a local shell with a destination of `localhost` or `127.0.0.1`, ZOC actually executes a command and redirects its input and output to the ZOC window.

By default `/usr/bin/login` will be started with parameters which automatically bypass the login prompt (using the currently logged on user).

If you prefer another default shell, you can configure it here. If necessary you can add the macro $(USERNAME) to the command if you need to pass the name of the currently logged on user as a parameter. Alternately you can provide a command in shebang format as the destination of a connection (e.g. connect to `#!/usr/bin/ssh secure.hogwarts.edu`), which will override this option.

The default value for this option is `/usr/bin/login -p -f $(USERNAME)`.

1.8.6.3 REXX ZocDeviceControl Command for Local Shell

`ZocDeviceControl` is a ZOC-REXX Command, which allows to perform functions that are specific to a commnunication method.
There are no available sub-commands for the `ZocDeviceControl` script command in relation to Local Shells:

1.8.7 UDP (Datagram)

UDP Datagram is a mechanism within local and public networks, where packets are exchanged between two IP Addresses without tracking the arrival of the transmission.

- UDP/Datagram Overview
- UDP/Datagram Options
- REXX ZocDevControl

1.8.7.1 UDP/Datagram Overview

The UDP/Datagram method lets you exchange UPD network packets with another computer. While most UDP based internet services like DNS or DHCP use binary formats which are not suitable for use with ZOC, there are other applications, which may make it necessary for ZOC to send or receive such packages, e.g. an upd-logger or an UDP short-message service.
To establish another IP address as a counterpart for UDP, use ZOC's Quick Connect function or create a host directory entry and provide the IP address of the other endpoint in the Connect to field, e.g. `192.168.1.10`.
After you click Connect ZOC will wait for UDP packets from that address and print their contents on the screen. The port number on which ZOC does listen for these packets is configured in ZOC' UDP-Options. Characters that you type will be sent to the other party as individual UDP packets.

1.8.7.2 UDP/Datagram Options

If you configure a connection for the UDP connection type, you can can use the following options:

Start session with Local Echo on
This option enables the Local Echo option on ZOC's session profile. Any characters you type and which will be sent to the other address, will also be printed on screen.

Port/Address where wackets will be received
A port number or a combination of local IP-address and port on which ZOC will listen incoming packets, e.g. `10024` or `192.168.1.100:10024`

1.8.7.3 REXX ZocDeviceControl for UDP

`ZocDeviceControl` is a ZOC-REXX Command, which allows to perform functions that are specific to a commnunication method.
There are no available sub-commands for the `ZocDeviceControl` script command in relation to UDP connections:

1.8.8 Named-Pipe

Within the Windows operating system, Named-Pipe is communication mechanism that allows two local programs (processes) to communicate with each other.
The following topics cover aspects in relation to using Named-Pipe within ZOC:

- Named-Pipe Overview
- Named-Pipe Options
- REXX ZocDevControl

1.8.8.1 Named-Pipe Communication Overview

The named pipe communication lets you access applications that provide a character based named pipe on your local machine or on a network server.
To connect to a named pipe, make a manual or host directory call and provide the name of the pipe in the connect-to field.
To connect to a pipe on your local machine use `\PIPE\<name>`, e.g. `\PIPE\OS2YOU` To connect to a pipe on a LAN server use `\\<server>\PIPE\<name>`, e.g. `\\ZAPHOD\PIPE\OS2YOU`

1.8.8.2 Named-Pipe Communication Options

If you set the pipe to accept incoming calls (ZOC File menu), the caller must connect to `\PIPE\ZOC`
Incoming pipes are only supported under Windows NT, XP, 2000, 2003

1.8.8.3 REXX ZocDeviceControl for Named-Pipe

`ZocDeviceControl` is a ZOC-REXX Command, which allows to perform functions that are specific to a commnunication method.
There are no available sub-commands for the `ZocDeviceControl` script command in relation to Named Pipe connections:

1.8.9 Rlogin

Rlogin is a communication mechanism that is similar to Telnet and is intended to use a remote shell over a network. It uses unencrypted transmission and has been mostly replaced by SSH.
The following topics cover aspects in relation to using Rlogin within ZOC:

- Rlogin Overview
- Options for Rlogin Connections
- REXX ZocDeviceControl

1.8.9.1 Rlogin Overview

The Rlogin communication method allows to access servers over a LAN or over the internet using the rlogin protocol. You can connect to host by using host's name or IP address in the connect-to field (e.g. in the host directory). The username is taken from the username setting. If it is empty, ZOC uses the value from the USERNAME environment variable, or, if USERNAME is unavailable, "ZOC" is used as the username. The escape character is hard coded to be the tilde character "~".

1.8.9.2 Rlogin Options

None.

1.8.9.3 REXX ZocDeviceControl for Rlogin

`ZocDeviceControl` is a ZOC-REXX Command, which allows to perform functions that are specific to a

commnunication method.
There are no available sub-commands for the `ZocDeviceControl` script command in relation to Rlogin connections.

1.9 Programming ZOC (REXX/DDE)

There are two methods of automating ZOC: REXX and DDE. Both are described in the topics below.

- ❑ REXX Programming Introduction
- ❑ REXX Language Elements (Variables, Decisions, Loops, etc.)
- ❑ ZOC Functions (by Category)
- ❑ ZOC Functions (Alphabetically)
- ❑ Online REXX Documentation

- ❑ DDE Dynamic-Data-Exchange Programming (Windows)
- ❑ AppleScript Programming (macOS)

1.9.1 Introduction to ZOC REXX

About REXX and ZOC The scripting language REXX has a long tradition as a scripting language, especially in the IBM world. It is a simple but powerful language which offers all language elements that are necessary for small and medium programming tasks (variables, decisions, loops, procedures, string manipulation, file I/O, etc.). There is even a REXX dialect (OOREXX) which offers object oriented language elements.
For scripting, ZOC uses this solid language as a foundation and extends it by adding specific functions for terminal emulation and communication. All these extensions have names starting with `Zoc`, e.g. `ZocConnect` or `ZocSend`.

Note: ZOC's implementation for the REXX interpreter for Windows and macOS is Regina REXX (http://regina-rexx.sourceforge.net/). If you prefer a different implementation (e.g. OOREXX), you can switch to that in Options→Program Settings→Special Files→Alternate REXX-DLL.

Learning ZOC REXX To get an impression of the language, you will find a quick tour through the very basics right below.
Once you completed this overview, continue with reading REXX Language Elements and ZOC-REXX Commands. These two topics will cover all language elements which are necessary to perform the majority of your tasks.
Beyond that, you will find the ZocScriptingSamples.zip archive in your My Documents→ZOC9 Files→REXX folder helpful. It contains a series of step-by-step examples with increasing complexity. On our website we then offer even more links to tutorials and samples.
If you want to use REXX on a very sophisticated level, you can look into the full ZOC REXX Reference (PDF), which you will also find in your My Documents→ZOC9 Files→REXX folder.

Hello World! ZOC REXX versions of the famous Hello World program looks like this:

```
/* REXX */
-- Display "Hello World" on the local screen
SAY "Hello World!"
```

```
/* REXX */
-- Send "Hello World" to the remote host
CALL ZocSend "Hello World!"
```

```
/* REXX */
-- Make a remote Unix host say "Hello World"
CALL ZocSend "echo 'Hello World!'^M"
```

General Considerations Usually all REXX programs begin with a comment, which REXX defines as any text which is written between `/*` and `*/`. This means that the first line in all REXX programs looks like this: `/* REXX */`. In addition to that, REXX also knows a form of comment which runs to the end of line. These comments start with `--` (see the above samples).
REXX knows native and non-native commands and functions. Examples for native commands are `SAY`, `IF`, `DO`, `END`. Some of the non-native (ZOC specific) commands are `ZocSend`, `ZocTimeout`, `ZocDisconnect`.
For easier reference, native language elements will written in uppercase, e.g. `CALL`, `SELECT`, `THEN`, `SAY`. Zoc extension commands are written in CamelCase, e.g. `ZocConnect`, `ZocSend`, etc. Finally names which are user defined (for example variables) are set in lowercase characters, `result= 10*userinput`.

How to Use Different Types of Commands

Native Commands

You can use native commands by writing the command name and optional arguments. Their placement is governed by the REXX language syntax and they create the basic structure of the program (the all uppercase words below are all native REXX commands):

```
IF rc=640 THEN DO
    SAY "Call Failed!"
  SIGNAL done
END
ELSE DO
  SAY "Success!"
END

EXIT
```

ZOC-Commands

ZOC-commands are executed by means of REXX's subroutine or function call mechanism. Essentially they come in two flavors:

For commands which do not return a result (or if you do not care about the result), you use the `CALL` command with the name of the ZOC-command and the required arguments separated by commas.

For commands which do return a value, you use an assignment and supply the argument list enclosed in brackets after the command name.

```
-- two commands without return values:
CALL ZocBeep 2
CALL ZocConnect "telnet.hogwarts.edu"
CALL ZocDownload "ZMODEM", "SOMEFOLDER"

-- this is a command which returns a value:
answer= ZocAsk("What is your name")
```

A Small Example Most of the time you will use REXX to log into a host and do things automatically. While simple logins sequences can be stored directly in the host directory without scripting (see Changing Host Directory Entries), there is a limit to what these can do.
The example below shows a more complex case. It calls a SSH host and checks for email by issuing the mail command on the host and looking at the result.

```
/*REXX*/

-- set variables for host login
hostip= "users.hogwarts.edu"
username= "harryp"
password= "alohomora"

-- make an SSH connection to the host
CALL ZocSetDevice "Secure Shell"
CALL ZocConnect username":"password"@"hostip

-- if login works, we should see the "Last login"
-- message within 10 seconds. (if ZocWait returns 640,
-- it means it did not appear in time)
CALL ZocTimeout 10
x= ZocWait("Last login")
IF x640 THEN DO
      -- we are in, so wait for a prompt
      CALL ZocWait "$"

      -- send the mail command
      CALL ZocSend "mail^M"

      -- skip the echo from our command
      CALL ZocWaitline
```

```
      -- wait for one more line of output and grab it
      CALL ZocWaitline
      themail= ZocLastline()

      -- logout and disconnect
      CALL ZocSend "exit^M"
      CALL ZocDelay 1
      CALL ZocDisconnect


      /* if the reply from mail was different from "No mail for ..."
         then display a message box to the user */
      IF LEFT(themail,7)"No mail" THEN DO
            CALL ZocMsgBox themail
      END
END
```

Note: More samples can be found in the ZocScriptingSamples.zip archive.

What now? Continue with the more detailed overview of REXX Language Elements.

1.9.2 Introduction to ZOC DDE Programming (Windows)

The Dynamic Data Exchange method (DDE) allows other Windows programs to execute ZOC-commands remotely. While REXX Programs must run under control of ZOC, it is possible to let separate applications (e.g. ones written in C++) send requests to ZOC for processing.
In this introduction it is assumed that the reader knows about the basic concepts of DDE. The topics below deal only with implementation specific details.
A more detailed description and sample applications can be downloaded from ftp://ftp.emtec.com/zoc/

Sessions/Topics

ZOC will accept only one session at a time. The application identifier is `ZOC` and ZOC responds to the topics `COMMUNICATION`, `COMM` and `COMM-DEBUG` (the difference is, that the latter logs DDE events in the ZOC window to help debugging). ZOC does not respond to general DDE-requests (ones that do not specify app or topic).
In addition to the application name `ZOC`, ZOC also responds to `ZOCn`, where n is a number that reflects the number of concurrently started ZOC instances. The first one is called `ZOC1`, the next one reacts to `ZOC2` etc.

Sending Commands

You can send commands to ZOC either via DDE_EXECUTE (data field) or DDE_REQUEST (item field). DDE_EXECUTE always responds with an DDE_ACK message with a numeric return code in the AppRc field. DDE_REQUEST always sends a DDE_DATA packet (even in case of error) with a result or with the string `##ERROR##` in the data field. The DDE_DATA packet must not be acknowledged. Under Windows the data handle must be freed by the receiver.

Serialization

Commands must be strictly sent on a command-reply basis. ZOC does not handle nested commands (e.g. sending another command while a ZocWait command is pending).

Command Format

DDE_EXECUTE and DDE_REQUEST accept the same form of commands, they only differ in the fact, that EXECUTE cannot be used to retrieve return values from functions. The commands must be zero terminated (C-style strings).
ZOC does not handle commands enclosed in square brackets (as recommended in the Microsoft DDE specification) and it does not handle multiple commands per message. Unicode is not supported.
A command can be sent in the form `<command> <arguments>` or `<command>(<arguments>)`. The arguments can be separated by blank or comma. However, the recommended style is with brackets and comma. String should be placed between quotation marks but it is legal to send them without if they do not contain special characters.
Here is a list of valid command strings. For backward compatibility it is legal to omit the letters ZOC at the beginning of the command words.

```
ZocBeep(2)
ZocMsgBox("Hello World!", 2)
ZocNotify('File "FOOBAR.DATA" not found!')

ZocRequest "How's life?" "Fine" "Not Bad"
Notify Error
Beep 2
```

List of Commands

Via DDE you can use the same set of commands/functions as in REXX. The list can be found in the ZOC-Command Reference in the appendix.

Example

The following code is a small example of how to start a DDE connection via Microsoft Access Visual Basic (VBA), to initialize the modem and check if it returns "OK".

```
ZocDDE = DDEInitiate("ZOC", "Comm-Debug")
DDEExecute ZocDDE, "ZocClearScreen"
DDEExecute ZocDDE, "ZocTimeout(10)"
DDEExecute ZocDDE, "ZocSend('ATZ^M')"
rc = DDERequest(ZocDDE, "ZocWait('OK')")
' or Excel 2003 and later:
'   rlist= DDERequest(ZocDDE, "ZocWait('OK')")
'   rc= rlist(1)
Debug.Print rc
DDETerminate ZocDDE
```

1.9.3 Introduction to ZOC AppleScript Programming (macOS)

While not being fully AppleScript aware, ZOC offers a simple interface to execute ZOC-Commands from within an AppleScript.
In order to prepare ZOC for the processing of those, the commands need to be enclosed between a `start conversation` and an `end conversation` command.

Calling ZOC Commands as Subroutines The ZOC commands themselves work as if they are issued from within ZOC's native scripting engine (REXX) and even the syntax is similar. To perform one of those, you simply put them into the script in function call syntax (the same syntax which you use to call a subroutine which resides inside an AppleScript).
A simple AppleScript using a ZOC function will look like this:

```
tell application "zoc9"
    -- prepare the current session in the first ZOC window
    -- to process upcoming ZOC commands.
    start conversation

    -- send 'ls -l<enter>' to the remote host
    ZocSend("ls -l^M")

    -- call a ZOC function for user interaction and store the result
    set reply to ZocRequest("Your preference?", "Apples", "Oranges")

    -- release the command processing
    end conversation
end tell
```

Please see the list of available commands for a description of parameters (this list is also available directly from ZOC's help menu). The syntax and examples there are tailored to ZOC's internal REXX scripting engine, but their use from within an AppleScript as shown above should be self evident.
Calling ZOC Commands as via 'perform' If you prefer to use a more AppleScript like syntax, you can also execute ZOC commands through the `perform` command. This command takes one direct parameter (the name of the ZOC command) and an array (in curly brackets) which contains the parameters (if any) to the respective command.
Using this variant is equivalent to the subroutine syntax. It merely provides a convenient alternative:

```
tell application "zoc9"
    -- prepare the current session in the first ZOC window
    -- to process upcoming ZOC commands.
    start conversation

    -- send 'ls -l<enter>' to the remote host
    perform "ZocSend" using { "ls -l^M" }

    -- call a ZOC function for user interaction and store the result
    set reply to perform "ZocRequest" using
         { "Your preference?", "Apples", "Oranges" }

    -- release the command processing
    end conversation
end tell
```

1.9.4 REXX Language Elements

Overview This topic offers an overview of the most commonly used parts of the native REXX language (a more basic introduction to ZOC programming can be found in REXX Programming).

The REXX language in combination with ZOC basically consists of three parts:
1. Basic REXX instructions (`CALL`, `IF`, `END`, etc.)
2. REXX built in functions. (`TIME()`, `SUBSTR()`, `STRIP()`, etc.)
3. ZOC Extensions (`ZocSend`, `ZocDownload`, `ZocWait`, etc.)
This topic covers the essential parts of points 1. and 2.
ZOC specific extension (3.) are listed in ZOC-REXX Commands/Functions.

If you look at the My Documents→ZOC Files→REXX folder on your computer, you will find a link to ZOC REXX Reference (PDF), document, which will cover all three points above in great depth. A link to other REXX resources. can also be found there.
Last but not least, there is a ZIP archive with ZOC scripting samples (ZocScriptingSamples.zip) in the folder mentioned above. Most of the topics which are presented here, are also demonstrated in the those samples.

Essential REXX Language Elements

Command Placement

In REXX comments can span multiple lines when they are placed between `/*` and `*/`. REXX also knows single line comments, which start with a `--` sequence and which automatically end when the line ends.

More than one command can be placed in one line by separating them by semicolons.

A command can be continued in the next line by placing an *extra* comma at the end of the line.

```
/* REXX
   Placement Sample */

SAY "Hello "; SAY "World"

CALL zoc "REQUEST", "What do you think?", ,
     "Do not know", "Do not care"

answer= ZOCRESULT()

-- now we're done
EXIT
```

Numbers and Arithmetics

Numbers can be used for calculation, counting etc. REXX uses numbers and the arithmetic in a pretty straightforward way:

```
/* REXX */
SAY 4*5+3*5 /* Output: 35 */
SAY 10/3    /* Output: 3.333333 */
SAY 10%3    /* Output: 3 (integral division) */
SAY 10//3   /* Output: 1 (division remainder) */
N= 4
SAY N       /* Output: 4 */
```

```
N= N+1
SAY N*3     /* Output: 15 */
```

Strings

REXX programs often process text in one way or another. While REXX inherently knows no types (values are treated as numbers or strings depending on the context), it is generally better to put all text strings in single or double quotes:

```
/* REXX */
SAY "Hello World!"
SAY "Joe's Bar"
SAY 'He suggested: "Use ZOC!"'
SAY 'She replied: "That''s my program!"'
```

You can assign strings to variables. Strings in quotes and string variables can simply be concatenated by writing them after one another, although sometimes the string concatenation operator `||` can be used for clarity or to resolve ambiguities.:

```
/* REXX */
w= "World"
ex= "!"
w2= w||ex   /* the || operator is actually required here: w||ex vs. wex */
hwe= "Hello "w2

-- say five times "Hello World!"
SAY hwe
SAY "Hello" w2
SAY "Hello "w2
SAY "Hello "||w2
SAY "Hello "||w||ex
```

Tracing

To find errors in your REXX program (or to watch the program execution) you can insert a `TRACE A` command into your REXX program (usually after the initial comment). With that, the REXX interpreter will display each instruction in the program as it executes it.

`TRACE I` will show very detailed tracing and `TRACE O` turns tracing off.

Conditionals

To express conditions in REXX programs, the following conditional operators are used:

Operator	Function	Example
=	equal (numbers)	`IF ret=640 THEN ...`

==	exactly equal (strings)	`IF input=="ZOC" THEN ...`
<>	not equal	`IF rc<>640 THEN ...`
\=	not equal (same as <>)	`IF rc\=640 THEN ...`
<	less	`IF val<10 THEN ...`
>	greater	`IF val>50 THEN ...`
<=	less or equal	`IF val<=100 THEN ...`
>=	greater or equal	`IF val>=100 THEN ...`
&	AND	`IF (i>=0) & (i<10) THEN ...`
|	OR	`IF (k=0) | (k=1) THEN ...`
\	NOT	`IF \(k=0) THEN ...`

Decisions
Decisions are made using the *IF <condition> THEN DO <commands> END ELSE DO <commands> END* sequence (see above for conditionals).
A typical program part using an *IF* command looks like this:

```
/* REXX */
IF rc=0 | rc=1 THEN DO /* rc 0  oder  rc 1 */
    SAY "ok"
END
ELSE DO
    SAY "failed"
END
```

The ELSE-part can be omitted if it is not needed. The keywords *DO* and *END* can be omitted if only one command is to be executed conditionally.:

```
/* REXX */
IF rc=0 | rc=1 THEN SAY "ok"
ELSE SAY "failed"
```

Loops
Loops are expressed in the following form:

```
DO <count> <commands> END
DO WHILE <condition> <commands> END
DO UNTIL <condition> <commands> END
DO <variable>=<start> TO <end> <commands> END
```

Example:

```
/* REXX */
DO 5
    SAY "Hello"
END

N= 100
DO WHILE n>0
    SAY n
    n= n-1
END

DO i=1 TO 10
    SAY i
END
```

Loops can be aborted from within using the `LEAVE` command. The `ITERATE` command will skip the rest of the loop body for the current iteration and will continue with the next loop iteration.

Jumps and Procedures (Subroutines)
Jump targets (labels) and subroutines are marked with a name and a colon. To jump to a label, use the `SIGNAL` command.

The `CALL` command is used to call a subroutine (which will return to the place it was called from via `RETURN`).

The following example demonstrates a jump and a call to a subroutine with argument passing:

```
/* REXX */

PULL n

IF n=0 THEN SIGNAL get_out

teny= 10/n
SAY "10/"n "is" teny
CALL square n

get_out:

EXIT

square:
    value= ARG(1)
    sqr= value*value
    SAY "Square of "||value||" is "||sqr
    RETURN
```

Function Calls
Functions are marked and called like procedures. The difference is that the can return a value to the

calling command:

```
/* REXX */

SAY "Enter base"
PULL b
SAY "Enter power"
PULL p

result= power(b,p)
SAY "The " p "th power of " b " is " result

EXIT

power:
    base= ARG(1)
    pow= ARG(2)
    res= 1
    DO I=1 TO pow
        res= res*base
    END
    RETURN res
```

External Calls

It is also possible to call an external script file as a subroutine and pick up a possible result. To do this, the name of the external script files needs to be placed in quotes. The called script then needs to be placed either in the ZOC program folder, in the ZOC Files folder, in the ZOC script files folder or in a folder which is listed in the PATH or REGINA_MACROS variable in the system environment (e.g. C:\1).

Alternately you can specify a full folder and file name for the file. Parameters are passed as usual (separated by commas).

If you have a scriptname or path in a REXX variable the regular call Syntax will not work though, but you can build a string which contains the complete command and pass it through the INTERPRET command (see the sample below).

```
/* REXX:test.zrx */

/* call external script in one of the ZOC folders */
CALL "sub.zrx" "Hello", "World"

/* call external script in an external folder */
CALL "C:\rexxlib\sub.zrx" "Hello", "World"

/* call external script as function */
x= "sub.zrx"(12,2)
SAY x

/* call external script via INTERPRET */
scriptfile= "Z:\test\zzz.zrx"
cmd= 'CALL "'||scriptfile||'"'
INTERPRET cmd
```

```
/* call external script as function via INTERPRET */
scriptfile= "sub.zrx"
cmd= 'x= "'||scriptfile'"(12,2)'
INTERPRET cmd
SAY x
```

The called script can pick up parameters through the built-in *ARG(n)*. function and return results via the *RETURN* command:

```
/* REXX:sub.zrx */
a= ARG(1)
b= ARG(2)
SAY a
SAY b
mult= a*b
RETURN mult
```

The PARSE command

PARSE is a powerful REXX command which can be used to split formatted strings in parts and assign the parts to variables. The syntax is *PARSE VALUE <string> WITH <variable>"<delimiter>" ...*

For example, if you have a string coordinate in the form *<index>: <pos-x>/<pos-y>* you can easily break it into parts via

```
PARSE VALUE coord WITH index": "posx"/"posy
```

Example:

```
/* REXX SAMPLE TO GET VALUES FROM A COMMA SEPARATED LIST */
x= "1, 2, 3, 4"

DO FOREVER
      IF x=="" THEN LEAVE

    PARSE VALUE x WITH val", "rest

      SAY val

      x= rest
END
```

Issuing ZOC Commands

ZOC commands are extensions to the REXX language which are implemented as procedures and functions (see ZOC-REXX Commands for a list).

ZOC commands that do not return a value (or if you are not interested in the return value) are called with the procedure call syntax:
CALL <zoc-command> <argument(s)>

ZOC-commands that do return a value are called with the function call syntax:
<result-var>= <zoc-command>(<argument(s)>)

Issuing Windows or macOS Shell Commands
Commands which are intended to be executed by the operating system (like deleting or renaming files), must be addressed to the operating system directly. This can either be done through REXX's *ADDRESS CMD* command or through ZOC's *ZocShell* command.

```
/* REXX */
ADDRESS CMD "cmd.exe /c DEL UPLOAD.TMP"
ZocShell "DEL UPLOAD.TMP"
```

One useful feature of *ADDRESS CMD* is to redirect the output to a stem (array) variable like this:

```
/* REXX */
ADDRESS CMD "CMD /C DIR" WITH OUTPUT STEM out.
SAY out.0 "lines of output"
DO i=1 TO out.0
      SAY out.i
END
```

Built-In Functions
The functions below can be used in assignments or in other commands that expect values, e.g. `b= ABS(a)` or `IF ABS(n)>10 THEN ...`.

Note: Only the essential functions and arguments are listed here, for a complete description please see The ZOC REXX Reference (PDF) in your My Documents→ZOC Files→REXX folder.

ABS(<value>)
Absolute value (remove sign), e.g. `n=ABS(t)`

ARG(<n>)
The n-th argument of a procedure/function.

COPIES(<str>,<repeat>)
Return <repeat> copies of <string>, e.g. `x=COPIES("-more- ", 10)`

C2D(<char>)
Obtain the decimal ASCII value of a character (Char-TO-Decimal), e.g. `n=C2D('A')` will set n to 65.

C2X(<char>)
Obtain the decimal ASCII value of one or more characters (Char-TO-heX), e.g. `n=C2X('AB')` will set n to 4142.

DATE("<style>")
Return the current date in different styles (one of B, D, E, M, N, O, S, U, W), e.g. `today=DATE("S")` will return `"20080201"` on February 1st, 2008).

D2C(<ascii>)
Create the character from a given ASCII value (Decimal-To-Character), e.g. `SAY D2C(65)` will print the letter A or `cr= D2C(13)` will assign the carriage return character to the variable CR.

FILESPEC(<part>, <filename>)
Cut a given part (one of "Drive", "Path", "Name") from a filename, e.g.
`dir=FILESPEC("Path", ofile)`

LEFT(<string>, <num>)
Return the first <num> characters from <string>, e.g. `SAY LEFT("BEERBOTTLE", 4)` will print BEER.

LENGTH(<string>)
Return the number of characters in a string.

LINEIN(<filename>)
Read the next line of text from a file (see FILE I/O).

LINEOUT(<filename>, <text>)
Write a next line of text to a file (see FILE I/O).

POS(<needle>, <haystack>)
Find the first string in the second and return the position or 0 if not found.

RIGHT(<string>, <num>)
Return the last <num> characters from <string>, e.g. `SAY RIGHT("BEERBOTTLE", 6)` will print BOTTLE.

STREAM(<filename>, ...)
Perform a file control operation (see the FILE I/O section below).

SUBSTR(<string>, <pos>[, <length>])
Return <length> characters from position <pos> in string. <length> can be omitted and will return the rest of the string.

STRIP(<string>)
Remove leading and trailing blanks from a string e.g. `str= STRIP(str)`

TIME("<style>")
Return the current time (style is one of C, H, L, M, N, S), e.g. SAY TIME("N") will show the time in HH:MM:SS format.

TRANSLATE(<string>)
Translate the characters in a string to upper case.

TRUNC(<n>, <m>)
Cut the value <n> so that it has <m> decimal places.

X2C(<ascii>)
Create the character(s) from a given hex value (heX-To-Character), e.g. `SAY X2C(41)` will print the letter A or `crlf= X2C(0D0A)` will assign the carriage return and line feed characters to the variable CRLF.

FILE I/O

Check if File Exists
```
IF STREAM(<filename>, "C", "QUERY EXISTS")\="" THEN ...
```

Open for Writing
```
CALL STREAM <filename>, "C", "OPEN WRITE"
```

Open for Reading
```
CALL STREAM <filename>, "C", "OPEN READ"
```

Write to File

```
CALL LINEOUT <filename>, <text>
```

Read from File

```
<variable>= LINEIN(<filename>)
```

Check End of File

```
IF STREAM(<filename>, "S")\="READY" THEN ...
```

Close File

```
CALL STREAM <filename>, "C", "CLOSE"
```

```
/* REXX FILE INPUT EXAMPLE */

file= "input.txt"

DO FOREVER
      ln= LINEIN(file)
      IF STREAM(file, "S")\="READY" THEN LEAVE

      /* process line of file (ln) here */

END
CALL STREAM file, "C", "CLOSE"
```

One useful trick to read a text file is the use of `ADDRESS CMD` and redirecting the output to a stem (array) variable like this:

```
/* REXX */
ADDRESS CMD "cmd.exe /c TYPE C:.csv" WITH OUTPUT STEM records.
SAY records.0 "records"
DO i=1 TO records.0
      SAY records.i
END
```

Please also check the FILEIO sample in SCRIPT\RXSAMPLE\TUTORIAL directory.

Parameters to REXX Scripts

It is possible to pass a parameter to a REXX scripts from the ZOC command line (or ZOC user buttons), e.g. `ZOC /RUN:script\test.zrx "/RUNARG:Hello, World"`
The parameter can be accessed via the ARG(1) function from within the REXX Script and if necessary be split using the `PARSE` command (it is recommended to subsequently use the STRIP function to remove leading and trailing blanks from the parameters).

```
/* PARAMETERS TO REXX */
p= ARG(1)
```

```
PARSE VALUE p WITH p1","p2
p1= STRIP(p1)
p2= STRIP(p2)
SAY "Parameters: ("||p||") ("||p1||") ("||p2||")"
```

Equivalents to BASIC Commands

ASC
D2C()

CHR$
C2D(), X2C()

CLOSE
CALL STREAM(<file>, "C", "CLOSE")

EOF
STREAM(<file>, "S")\="READY"

FOR ... NEXT
DO n=<first> TO <last> ... END

INSTR$
POS()

MID$
SUBSTR()

LEN
LENGTH()

OPEN
STREAM(<file>, "C", "OPEN ...")

INPUT
PULL, ZocAsk()

INPUT#
LINEIN()

PRINT
SAY, ZocWrite, ZocWriteln

PRINT#
LINEOUT

1.9.5 ZOC Functions by Category

Below follows a List of ZOC REXX Functions/Commands sorted by category. Alternately an alphabetic list is found in the alphabetic ZOC-Command Reference in the Appendix. Additionally you will find a description of basic REXX syntax and operations (like variables, loops, etc.) in REXX Language Elements.

Connecting to Hosts and Sending/Receiving of Data

- ZocConnect
- ZocConnectHostdirEntry
- ZocDisconnect
- ZocDownload
- ZocLastLine
- ZocPing
- ZocReceiveBuf
- ZocRespond
- ZocSend
- ZocSendEmulationKey
- ZocSendRaw
- ZocSetAutoAccept
- ZocTimeout
- ZocUpload
- ZocWait
- ZocWaitForSeq
- ZocWaitIdle
- ZocWaitLine
- ZocWaitMux

Local Input/Output

- ZocAsk
- ZocAskPassword
- ZocAskFilename
- ZocAskFilenames
- ZocAskFoldername
- ZocBeep
- ZocClearScreen.
- ZocDialog.
- ZocGetScreen
- ZocKeyboard
- ZocMessageBox
- ZocNotify
- ZocPlaySound
- ZocRequest
- ZocRequestList
- ZocSuppressOutput
- ZocWrite

- ZocWriteln
- ZocWindowState

Settings and Options

- ZocGetProgramOption
- ZocGetSessionOption
- ZocGetHostEntry
- ZocLoadKeyboardProfile
- ZocLoadSessionProfile
- ZocLoadTranslationProfile
- ZocSaveSessionProfile
- ZocSetDevice
- ZocSetDeviceOpts
- ZocSetEmulation
- ZocSetHostEntry
- ZocSetProgramOption
- ZocSetSessionOption
- ZocSetUnattended

Local Program Execution and File Manipulation

- ZocFilename
- ZocFileCopy
- ZocFileDelete
- ZocFileRename
- ZocListFiles
- ZocShell
- ZocShellExec
- ZocShellOpen

Performing ZOC Operations (Commands, Menu, Tabs)

- ZocCommand
- ZocDelay
- ZocGetInfo
- ZocLogname
- ZocLogging
- ZocMenuEvent

- ZocSessionTab
- ZocTerminate

Miscellaneous Functions

- ZocClipboard
- ZocMath
- ZocGlobalStore
- ZocRegistry
- ZocString

Advanced, Subtle and Obscure Functions

- ZocCtrlString
- ZocDdeClient
- ZocDoString
- ZocDeviceControl
- ZocEventSemaphore
- ZocSetAuditLogname
- ZocSetMode
- ZocSetTimer
- ZocSynctime

1.9.6 Alphabetic List of ZOC REXX-Functions

An alphabetic list of ZOC REXX functions is found in the ZOC-Command Reference in the Appendix.

1.10 Updates, Author, Email, Orders, ...

1.10.1 Contacting EmTec

Ordering
If you have a question about ordering, please visit http://www.emtec.com/common/order.html for all ordering details or select 'Order-Form' from the ZOC Help menu or from the ZOC Program Group in your Windows Start Menu. There you will find the list of distributors and ways to contact them.

Support
If you have a question, suggestion, or if something does not work, we are really interested to help. But please (pretty please with sugar on top), before writing an support email, check out Problems and Questions.
If you do not find the answer there, feel contact support via our website:
http://www.emtec.com/common/support.html

Contact
If you have complex technical problems, need development support or custom programming, have ideas or bug reports, need a site license, want to send a general comment or Christmas presents please contact EmTec via: http://www.emtec.com/common/contact.html or
EmTec Innovative Software
Markus Schmidt
Kirchenweg 14
90419 Nuernberg
- Germany -

1.10.2 Finding Updates

http://www.emtec.com/zoc/index.html
http://www.bmtmicro.com
ftp://ftp.emtec.com/zoc

1.10.3 Copyrights and Trademarks

This software also contains source code and program parts from the following sources (which is used in accordance with their respective software license):
Copyrights

OpenSSL

Copyright (C) 1997 Eric Young (eay@cryptsoft.com) and the (C) The OpenSSL project. See http://www.openssl.org/source/license.html

OpenSSH

Parts of the SSH code are taken from the OpenSSH project and from the original SSH source code under BSD-licenses and some parts in the public domain. Holders of Copyright are Tatu Ylonen, Markus Friedl, Dug Song, Theo de Raadt, Niels Provos and other contributors. All rights reserved. See LICENSE at https://www.openssh.com/openbsd.html

LIBFIDO2

LIBFIDO2 is provided under a BSD-2-Clause license. Copyright (c) 2018-2023 Yubico AB. All rights reserved. https://github.com/Yubico/libfido2/blob/main/LICENSE

LIBSSH2

Use of LIBSSH2 is allowed by retaining the following copyrights: Copyright (C) Sara Golemon, Mikhail Gusarov, The Written Word, Inc., Eli Fant, Daniel Stenberg, Simon Josefsson, Markus Friedl, Microsoft Corp. All rights reserved. See: https://libssh2.org/license.html

REGINA/ZOC REXX

The ZOC Rexx processor is based on the REGINA REXX interpreter (C) 2008 Mark Hessling (http://regina-rexx.sourceforge.net/).

In accordance with its LGPL license (see ZOCDLL-COPYING.TXT) you can obtain the full source code for ZOCREXX.DLL library via a request to our support.

OOREXX

The OOREXX system is (C) by http://www.oorexx.org and provided under the OOREXX License: http://www.oorexx.org./license.html

InfoZip/Install

Copyright (C) 1990-1992 Mark Adler, Richard B. Wales, Jean-loup Gailly, Kai Uwe Rommel and Igor Mandrichenko
explode.c -- Not copyrighted 1992 by Mark Adler
inflate.c -- Not copyrighted 1992-94 by Mark Adler

TN3270

ZOC TN3270 is based on an ancient release of the 3270 emulator by Georgia Tech with the following copyright: Copyright 1989 by Georgia Tech Research Corporation, Atlanta, GA 30332. All Rights Reserved. GTRC hereby grants public use of this software.

Some parts:
Copyright (c) 1993-2009, Paul Mattes.
Copyright (c) 1990, Jeff Sparkes.
ILINKh(ttp://x3270.bgp.nu/x3270-man.html#Copyrights)

CRC Routines

UPDCRC macro derived from article Copyright (C) 1986 Stephen Satchell.
CRC-Table copyright (C) 1986 Gary S. Brown.

Rlogin

Copyright (C) 1983 Regents of the University of California.

Tapi

Copyright (C) 1995 Microsoft Corporation. All Rights Reserved.

Zlib (gzip)

Copyright (C) 1995-1996 Jean-loup Gailly and Mark Adler.

Zmodem

Original Version by Chuck Forsberg, Omen Technology Inc.

Trademarks

ZOC, EmTec
ZOC and EmTec are registered trademarks of Markus Schmidt

Microsoft, Microsoft Windows
Microsoft and Microsoft Windows are registered trademarks of Microsoft Corporation

VT52, VT100, VT102, VT220, VT320, VT420, VT520
VT52, VT100, VT102, VT220, VT320, VT420 and VT520 are trademarks of the Digital Equipment Corporation

IBM, IBM3270, IBM5250, AS/400, OS/2, iSeries, eServer
IBM, IBM3270, IBM5250, AS/400, OS/2, iSeries and eServer are trademarks of IBM Corporation

Apple, Macintosh, macOS
Apple, Macintosh, macOS are trademarks of Apple Inc.

1.11 APPENDIX

- Table of Control Characters and Action Codes
- Extended Escape-Sequces
- VT102/VT220 Special Keys on the PC Keyboard
- 3270 Special Keys on the PC Keyboard
- Key Names for ZocSendEmuKey/^KEY
- REXX Language Elements
- ZOC REXX-Commands/Functions
- Features You May Have Missed
- Common Questions (How-To Guide)
- Common Problems and Questions (Trouble Shooting Guide)

1.11.1 Special Codes (Ctrl-Characters, Placeholder- and Action Codes)

- Control Codes
- Placeholder Codes

Sometimes it is necessary to send characters that are not available from the keyboard or that cannot be entered into a field because the operating system uses it to move the cursor (like the ESC or Tab or the Enter key). The list below shows what control codes to enter in an action-field (e.g. `ZOC^M`), in order to perform the equivalent such special key (e.g. to send `ZOC`<ENTER>).
If need to send those codes from the terminal window instead of mapping them to a button or, for example in a case whwere you need to send `^]` to wake up a Cisco device, please see the list of key equivalents for control codes in sending of control codes.
ZOC also offers special placeholder codes which allows you define some filenames (e.g. for Logfiles) in a way that they can conain date and time values.

Control Codes
The codes listed here can be added to actions in order to send equivalents of keys that can not be entered into dialog fields otherwise (e.g. when entering text to remap keys, etc.). For example, in order to remap a F-key so that it will send the text `exit` followed by an Enter key, you need to specify the text to send as `exit^M`.

Control Code	Description	Emulation Key Equivalent	TN3270 Key Equivalent
^M	Carriage-Return (CD, hex 0D)	Enter	Newline

^[	Escape \E \033	Esc	
^H	Backspace (hex 08)	Backspace	Backspace
^I	Tabulation (hex 09)	Tab	Tab
^J	Line-Feed (LF, hex 0A)	.	Newline
^Z	Suspend process (hex 1A)	Ctrl+Z	Transmit/Send Data
Other ^A ... ^Z	Combination with Ctrl-key	Ctrl+A ... Ctrl+Z	.
^~	Ctrl+^ (dec 30)	.	.
^[[	DEC Control Squenze Introducer CSI	.	.
^^	^ character	.	.
^?	DEL-character (dec 127 or hex 7Fh)	.	.
^!	Send a modem break signal or equivalent	.	.
^°	125 ms pause between the previous and next character	.	.
^(xx)	Any character with code of hex xx e.g. `^(7F)` for DEL	.	.

. **Placeholder Codes**
The table below lists place holders that can be used in some ZOC fields to insert date, time or connection related values. They can be used in all places that support normal control codes and also in Logfile names (Options→Session Profile→Logging).

Placeholder Code	Replacement Value
^&	Replaced by the password from the host directory entry that is currently online (for safety reasons this works only once per session).
^%	Replaced by the username from the host directory entry that is currently online
^+	Replaced by the name of the remote host online
^1	Replaced by current day of month
^2	Replaced by current month
^3	Replaced by current year
^4	Replaced by current hour
^5	Replaced by current minute
^6	Replaced by current second
^7	Replaced by local computer name
^8	The instance number of the currently open ZOC window If the window has multiple tabs, a counter for the tabs will also be added. This code is mostly used with log filenames to prevent multiple instances from using the same file.
^9	ZOC version and operating system, e.g. ZOC8.08.7-WIN10

1.11.2 Extended Escape-Sequences

The following escape-sequences extend the normal range of sequences (the normal sequences define the behavior of ZOC depending on the chosen each emulation).
Their main purpose of the extended sequences is to allow a host to control specific features or invoke specific actions through ZOC.
The sequences typically start with `ESC ]` (*hex 1B 5D*) or `ESC [` (*hex 1B 5B*), followed by a variable part and a termination character like `^G` (*hex 07*) or `CR` (*hex 0D*).

Other characters in the syntactic samples below denote themselves, except for spaces (which should be ignored) and text in brackets (which is a placeholder for a variable part of your own choice).
For example, a valid implementation of the sequence *ESC] 0 ; <title> ^G* is the byte series *hex 1B 5D 30 3B 5A 4F 43 07* (set window title to "ZOC").
On most Linux systems, you could emit the sequence from the example above via `echo -e "\e]0;ZOC\a"`. On programming languages that use C-like strings (C, C++, Python, Perl, PHP, ...), you can use something like `print "\033]0;ZOC\007";`.

Change Window Title:* *ESC] 0 ; <title> ^G
The sequence allows the remote program to change the title for the window (or tab) in which the session runs.

Change Window Title (alternate):* *ESC [| $ t = <title> CR
The sequence is an alternate version of the above and also allows the remote program to change the title of the window.

Clear Screen and Reset Emulation:* *ESC] R
A byte sequence of *hex 1B 5D 52* will perform a clear screen and emulation reset (this is identical to the user choosing Clear Screen from the View menu).

Change Window Title:* *ESC] 52 ; c ; <base64-data> ^G
The OSC52 sequence allows the remote program to put text to the clipboard according to the OSC 52 protocol (requires activation of the 'Start Local Command', see below).

Start Local Command:* *ESC [| <command> CR
A host can send this sequence to perform a command locally through ZOC.

Important: Since this is a potential security risk, the sequence is ignored, unless the option to allow it is set in the Options→Session Profile→Advanced. Alternately, the sequence is also processed, if a file named `remoterun.flg` exists in the ZOC program folder (Windows) or if a file named `.zoc9_remoterunflag` exists in the user's home folder (macOS).

The command is passed to Windows through the CreateProcess API and it can contain optional parameters, e.g. `notepad.exe newfile.txt` or through a cmd.exe `C:\Windows\system32\cmd.exe /c dir >out.txt`. Under macOS you can run programs directly (/bin/touch /tmp/didit.txt). Tasks which require shell features (e.g. output redirection), will need to run through a shell e.g. `/bin/bash -c ls ~/Downloads >/tmp/out.dir` or through shell as a shell script `/bin/bash /Users/you/thetask.sh`

If the command parameter starts with `http://`, `https://` or `ftp://`, the URL will be passed to user's the default web browser.

If the command starts with the text `%SHELLOPEN%` followed by a space character and a file name, the file will be opened through the operating system using the user's chosen default program, e.g. `%SHELLOPEN% C:\ZOC\piechart.jpg`

There are also some placeholders: *DOWNLOADDIR*, *LASTDOWNLOADEDFILE*, *LASTSPOOLFILE*. When these appear either between percent-signs (e.g. `%LASTDOWNLOADEDFILE%`) or in brackets after a dollar sign (e.g `$(LASTSPOOLFILE)`), they will be replaced by their respective values before executing the command. A typical sample could be a host sending a report to the user via Zmodem and then opening it via `%SHELLOPEN% "%DOWNLOADDIR%\thereport.pdf"` or `%SHELLOPEN% "%LASTDOWNLOADEDFILE%"`

NetTerm Compatibility:* *ESC [ ] <parameter> ESC [<n> *
ZOC supports a range of sequences from NetTerm to perform operations on the local computer that runs the terminal.

The parameter <n> is a decimal number which determines the function to perform. The meaning of the <parameter> thus depends on <n> as described below:

0 - URL to open in the local browser, e.g. `ESC [ ] https://www.emtec.com/ ESC [ 0 *`.
1 - Command to perform locally (same as the 'start local command' sequence above).
2 - Perform a ZOC action (this is an action string, like actions which you will find if you look at a ZOC session profile file using an editor, when you go to the user button section).
6 - Directory to use as the download directory.
7 - Directory to use as the upload directory.
8 - Name and path of file to transfer when the next upload is invoked.
11 - Text to write to the local clipboard.
12 - Full path and name of a file to open using the associated default program for that file's type. The placeholder `%LASTDOWNLOADEDFILE%` can be used to refer to the last file that was received through a download.
17 - The text `NOWAIT`. In this case, the last downloaded file will be opened on the local computer (using the default program for that file's type).
17 - The text `PRINT`. The last downloaded file will be sent to the printer that is configured in Program Settings→Printer.

17 - The text `DELETE`. The last downloaded file will be deleted.

Example: `ESC [ ] C:\DOWNLOADS ESC [ 6 *` (*`hex 1B 5B 5D 43 3A 5C 44 4F 57 4E 4C 4F 41 44 53 1B 5B 36 2A`* will subsequently (for this session only) store downloaded files in the folder C:\DOWNLOADS\.

Example: `ESC [ ] C:\DOWNLOADS\REPORT.PDF ESC [ 12 *` will open the file REPORT.PDF on the local computer, using the default viewer for PDF files.

Example: `ESC [ ] ^RUN=postprocess.zrx ESC [ 2 *` will run the postprocess.zrx REXX script.

Example: `ESC [ ] NOWAIT ESC [ 12 *` will open the last downloaded file (e.g. MYREPORT.PDF) on the local computer, using the default viewer for PDF files.

Important: These sequences will only be processed, if the `remoterun.flg` is present on the system or if they are enabled under Options→Session Profile→Advanced (see 'start local command' above). Also, if you want to debug your use of these sequences, you can enable hex mode or ascii tracing in Options→Session Profile→Trace/Debug.

1.11.3 Keyboard Topics

- Menu Shortcuts
- Default Key-Mappings for the xterm/VT220/VT420/Linux Emulations
- Default Key-Mappings for the TN3270 Emulation
- Default Key-Mappings for the Wyse/TVI Emulations
- Default Key-Mappings for the TN5250 Emulation
- Default Key-Mappings for the Tandem 6350 Emulation

- Key-Names for User Defined Key-Remapping
- Creating User Defined Keyboard Mappings
- Symbols used to Send Control Codes

Menu Shortcuts

Key	Function
Alt+B	View → Show Data Stream Viewer
Alt+C	View → Local Typing
Alt+D	File → Host Directory
Alt+E	View → Editor
Alt+F	View → Find in Scrollback
Alt+H	File → Disconnect
Alt+K	View → Clear Scroll Buffer
Alt+L	Enable/Disable Log to File
Shift+Alt+L	Logging → Set Logfile Name
Alt+O	File → Quick Connection
Alt+P	Logging → Log To Printer
Shift+Alt+R	File → Print Screen
Alt+R	File → Reconnect
Shift+Alt+T	File → New Tab
Shift+Alt+W	File → Close Tab or Window
Alt+X	End Program
Alt+W	Close Window or Tab
Alt Y	View → Clear Screen
Ctrl+Tab	View → Next Tab
Ctrl+Shift+Tab	View → Previous Tab
Shift+Esc	View → Session-Thumbnails
Shift+PgUp	View → Scroll Back
Alt+PgUp	View → Scroll Back as Window
Ctrl+PgUp	Transfer → Upload
Ctrl+Shift+PgUp	Transfer → Send text file
Ctrl+PgDn	Transfer → Download
Shift+Insert	Edit → Paste
Alt+Insert	Edit → Paste (no line breaks)
Alt .	Options → Session Profile
Alt ,	Options → Program Settings
Alt =	View → Enable/Disable Split Chat
Alt +	Script → Start REXX Script
Alt -	Script → Stop REXX Script
Alt *	Script → Edit REXX Script
Alt End	File → Send Break

Sending Control Codes

Key Combination	Control Character
Ctrl+Letter (a-z)	Send ^A ... ^Z (hex 01 ... hex 1A)
Ctrl+Shift+Letter (A-Z)	Send ^A ... ^Z (hex 01 ... hex 1A)
Ctrl+@ or Ctrl+Space	Send ^@ (hex 00)
Ctrl+[or Ctrl+3	Send ^[(hex 1B)
Ctrl+\ or Ctrl+4	Send ^\ (hex 1C)
Ctrl+] or Ctrl+5	Send ^[(hex 1D)
Ctrl+^ or Ctrl+~ or Ctrl+6	Send ^^ (hex 1E)
Ctrl+[or Ctrl+? or Ctrl+7	Send ^_ (hex 1F)
Ctrl+8	Send DEL (hex 7F)

1.11.4 VT102/VT220/Linux/Xterm Keys on the Local Keyboard

The list of keys below shows how the emulation's keys are mapped to the your local keyboard. This happens automatically when you choose to use the emulation for a connection

However, any of these keys can also be mapped to any other key-combination on the local keyboard via the Options→Keyboard Profiles. They can also be mapped to user-defined clickable screen buttons via (Options→Session Profile→User Buttons). Both functions use names for the keys listed in the table of emulation dependent key names.

Main Keyboard Keys:

Xterm/VT220/VT420/VT520/VT102/Linux Key-Mappings	Equivalent on the PC Keyboard
[F1] ... [F4]	not available
[PF1] ... [PF4]	[F1] ... [F4] sup class="helpsup"[1]/sup
[F5] ... [F10]	[F5] ... [F10] sup class="helpsup"[1]/sup
[F11]/[F12]	[F11]/[F12] or Shift+[F1]/[F2] (*)
[F13] ... [F20]	Shift+[F3] ... [F10] sup class="helpsup"[1]/sup
[Insert Here]	[Ins] Shift+[Home]
[Remove]	[Del]
[Select]	[Home]
[Find]	[End]
[Prev Screen]	[PageUp]
[Next Screen]	[PageDown]

sup class="helpsup"[1]/sup Some servers use variants of the F-Keys. In fact especially for the Xterm emulation the mappings for the function keys are quite a mess (see the internet link below). If you have trouble with the built in codes, try to create a keyboard profile (Options menu) and map the F-Key with values from the following table: Common Codes for F-Key Mappings.

Numeric/Auxiliary Keypad:

VT102 Aux Keypad	PC Numeric Keypad (Num Lock Off)
[PF1]	[F1]
[PF2]	[F2]
[PF3]	[F3]
[PF4]	[F4]
[0] ... [9]	[0] ... [9]
[Enter]	[Enter]
[.]	[./Del]
[-]	[-]
[,]	[+]

1.11.5 TN3270 Key-Mappings

The list of keys below shows how the emulation's keys are mapped to the your local keyboard. This happens automatically when you choose to use the emulation for a connection and is subject to some configuration options of the TN3270 emulation via Options→Session Profile→Emulation→TN3270.

However, any of these keys can also be mapped to any other key-combination on the local keyboard via the Options→Keyboard Profiles. They can also be mapped to user-defined clickable screen buttons via (Options→Session Profile→User Buttons). Both functions use names for the keys listed in the table of emulation dependent key names.

TN3270 Default Keyboard Mapping:

3270 Key	PC Key
[Newline]	[Return] sup class="helpsup"[1]/sup
[Enter/Transmit]	[Enter] (numeric keypad)
	Shift+[Return] sup class="helpsup"[1]/sup
	[right Ctrl Key]
[Reset]	[Esc]
	[left Ctrl-key] (Windows only)
[Insert]	[Ins] (Windows only)
	Shift+[Backspace]
	Shift+[Del]
	Shift+[Home]
[Attn]	[Ctrl+C]
	Shift+[Pause] (Windows only)
[PA1] ... [PA3]	Alt+[1] ... [3]
[EraseEOF]	[End]
[Clear]	Shift+[Esc]
	[Pause] (Windows only)
	[Clear] (macOS only)
[EraseInput]	Alt+[Del]
[PF1] ... [PF12]	[F1] ... [F12] .
[PF13] ... [PF24]	Shift+[F1] ... [F12] .
[PF7]/[PF8]	[PageUp] / [PageDown]
[Left2]	Shift+[Left]
[Right2]	Shift+[Right]

Special ZOC Functions

The mappings below perform functions, which are not found on an original TN3270 terminal but which may be especially useful in the context of a terminal emulator, e.g. ZOC offers an undo function like in a local editor or on a text processing program.

ZOC 3270 Function	PC Key	Cor
[HomeCmd]	Alt+[Home]	Jum
[MovePrevWord]	Alt+[Left]	&nb
[MoveNextWord]	Alt+[Right]	&nb
[DeleteNextWord]	Ctrl+Del	&nb
[MoveEOF]	Ctrl+[End]	Mov
[MoveBOL]	Alt+[Enter]	Mov
[SelectAll]	Ctrl+A	&nb
[EraseEOF]	Ctrl+K	Era
[EraseField]	Ctrl+L	Era
[SaveCursorPos]	Ctrl+S	Sav
[RestoreCursorPos]	Ctrl+R	Jum

[Undo]	Ctrl+Z	Und
[Redo]	Ctrl+Shift+Z	Rev

sup class="helpsup"[1]/sup The functions of Return and Shift+Return (also Alt+Return) can be swapped in the settings of the TN3270 emulation via Options→Session Profile→Emulation→TN3270. The combination Cmd+[Return] (macOS only) is unaffected by the swap option there.

1.11.6 Wyse/Televideo Key-Mappings

The list of keys below shows how the emulation's keys are mapped to the your local keyboard. This happens automatically when you choose to use the emulation for a connection

However, any of these keys can also be mapped to any other key-combination on the local keyboard via the Options→Keyboard Profiles. They can also be mapped to user-defined clickable screen buttons via (Options→Session Profile→User Buttons). Both functions use names for the keys listed in the table of emulation dependent key names.

Main Keyboard Keys:

Wyse/Televideo Key-Mappings	Equivalent on the PC Keyboard
[F1] ... [PF12]	[F1] ... [F12]
[F13] ... [F24]	[Shift+F1] ... [Shift+F12]
[InsChar]	[Shift+Ins]
[DelChar]	[Shift+Del]
[Transmit]	[Alt+Enter]

1.11.7 TN5250 Key-Mappings

The list of keys below shows how the emulation's keys are mapped to the your local keyboard. This happens automatically when you choose to use the emulation for a connection and is subject to some configuration options of the TN5250 emulation via Options→Session Profile→Emulation→TN5250.

However, any of these keys can also be mapped to any other key-combination on the local keyboard via the Options→Keyboard Profiles. They can also be mapped to user-defined clickable screen buttons via (Options→Session Profile→User Buttons). Both functions use names for the keys listed in the table of emulation dependent key names.

TN5250 Default Keyboard Mappings

5250 Key	PC Key
[PF1] ... [PF12]	[F1] ... [F12]
[PF13] ... [PF24]	[Shift+F1] ... [Shift+F12]
[Newline]	[Return] (*)
[Enter/Transmit]	Shift+[Return] (*) Alt+[Return] (*) [right Ctrl-key] (Windows only) Cmd+[Enter](*) (macOS only)
[Field Exit]	[Enter] (numeric keypad)
[Reset]	[Esc] [Left Ctrl-key] (Windows only)
[Ins]	[Ins] (Windows only)

	Shift+[Backspace]
	Shift+[Del]
	Shift+[Home]
[EraseEOF]	Alt+[End]

(*) The functions of Return and Shift+Return (also Alt+Return) can be swapped in the settings of the TN5250 emulation. The combination Cmd+[Return] (macOS only) is unaffected by the swap option.

Accessing 5250 Keys via Esc-Combination

Some 5250 emulators allow access to special keys via a Esc key combination.
ZOC however uses Esc for the Reset function, but the Esc combinations are available if you remap the Esc key in Options→Keyboard Profiles as the emulation dependent key named `EscapeMenu`.
With that configuration you will be able to simulate most 5250 keys by typing a two-key sequence as shown below (e.g. `Esc A` will simulate the Attention key.).
However, rather than using two keystrokes, it will probably be more convenient to remap the keys you need directly to a PC keyboard key in the Options→Keyboard Profiles, mapping the PC keys to simulating a key based on Emulation Dependent Key Names.

1 = F1
2 = F2
3 = F3
4 = F4
5 = F5
6 = F6
7 = F7
8 = F8
9 = F9
0 = F10
- = F11
= = F12
! = F13
@ = F14
# = F15
$ = F16
% = F17
^ = F18
& = F19
* = F20
(= F21
: = F22
_ = F23
+ = F24
A = Attention
D = Duplicate
H = Help
I = Insert
L = Refresh
M = Fieldminus
P = Print
R = Reset
S = Sysreq
X = Fieldexit

1.11.8 Tandem 6530 Key-Mappings

The list of keys below shows how the emulation's keys are mapped to the your local keyboard. This happens automatically when you choose to use the emulation for a connection.

However, any of these keys can also be mapped to any other key-combination on the local keyboard via the Options→Keyboard Profiles. They can also be mapped to user-defined clickable screen buttons via (Options→Session Profile→User Buttons). Both functions use names for the keys listed in the table of emulation dependent key names.

Tandem 6530 Key	PC Key
[F1] ... [F12]	[F1] ... [F12]
[F13] ... [F16]	[Alt+F3] ... [Alt+F6]
[Enter]	[Return]
	Shift+[Return]
	Ctrl+[Return]

1.11.9 Names of Emulation Keys for User-Defined Actions

All emulations within ZOC have default key mappings built in, where they automatically map emulation specific keys (e.g. the VT220 'Next' key) to an equivalent (or suitable) PC or Mac Key. See Keyboard Default Mappings.
All keys of these emulations can also be used in a different context, e.g., for user defined key mappings in Options→Keyboard Profiles, for the REXX function ZocSendEmulationKey, or for Options→Session Profile→User Buttons. This is done via predefined key designations (names) that are available while the corresponding emulation is activated.

Emulation	Available Key Names
Xterm	Esc, Up, Down, Right, Left, Home, End, Insert, Enter, Return, Backspace, Del, Delete, PgUp (or Prev), PgDown (or Next), Home, End, Tab, BackTab, F1, F2, ..., F20 (VT220 style), XF1, XF2, ..., XF24 (Xterm style), RF1, RF2, ..., RF24 (rxvt style), ModUp, ModDown, ModRight, ModLeft, ModHome, ModEnd.
VT220/VT320/VT420	Esc, Pf1, Pf2, Pf3, Pf4, F5, F6, ... F20, Help (same as F15), Do (same as F16), Find, Insert, Remove, Select, Prev, Next, Aux0, Aux1, Aux2, Aux3, Aux4, Aux5, Aux6, Aux7, Aux8, Aux9, AuxDot, AuxMinus, AuxPlus, AuxEnter, Enter, Crlf, Tab, Backspace, Del, Up, Down, Right, Left, ModUp, ModDown, ModRight, ModLeft.
TN3270	Insert, Enter, Tab, BackTab, Reset, Home, Left, Left2, Right, Right2, Up, Down, NewLine, SysReq, Attn, Pa1, Pa2, Pa3, Clear, Backspace, BackDel, Delete, EraseEOF, MoveEOF, MonoCase, Dup, FieldMark, EraseInput, CursorSelect, PF1, PF2, ..., PF24. ZOC Extensions: Undo, Redo, HomeCmd, LowerHome, HomeAny, MoveNextWord, MovePrevWord, DeleteNextWord, MoveBOL, EraseField, Lightpen, CharNot (¬), MoveTo:r/c (r/c= row/column).
Linux	Esc, Up, Down, Right, Left, Home, End, Insert, ModUp, ModDown, ModRight, ModLeft, Enter, Return, Backspace, Del, Delete, PgUp (or Prev), PgDown (or Next), Tab, BackTab, F1, F2, ..., F20.

TN5250	Attn, Backspace, BackDel, BackTab, Delete, Duplicate, EraseEOF, End, Enter, FieldExit, FieldMinus, FieldPlus, Home, Help, Insert, Newline, RollDown, RollUp, Refresh, Reset, SysReq, Tab, F1, F2, ..., F24.
Wyse/Televideo	Enter, Return, Transmit, Linefeed, Backspace, Esc, Up, Down, Right, Left, PgUp, PgDown, Home, End, Insert, InsChar, Delete, Del, Tab, BackTab, FieldTab, F1, F2, ..., F16.
Sun CDE	Same as VT220 but use KF1-KF4 instead of PF1-PF4.
ANSI BBS	Esc, Up, Down, Right, Left, Home, End, Insert, Enter, Return, Backspace, Del.
ANSI SCO	Esc, Up, Down, Right, Left, Home, End, Insert, Enter, Return, Backspace, Del, Delete, PgUp, PgDown, F1, F2, ..., F24.
Tandem 6530	Enter, CtrlEnter, ShiftEnter, Home, CtrlHome, End, Tab, BackTab, Insert, Delete, BackSpace, Up, Down, Left, Right, PgDown, PgUp, InsertLine, DeleteLine, F1, F2, ..., F16.

Example: when the VT220 emulation is active, you can remap Ctrl+I as the Insert key by going to Options→Keyboard Profiles. Then select Ctrl+I, click the Mapping Assistant button, select Send a single emulation specific key and enter the text `Insert` in the Send key field.
You can also invoke these keys through Options→Session Profile→User Buttons and the REXX command `ZocSendEmulationKey`, e.g. in a REXX scripts you can use the command `Call ZocSendEmulationKey "Insert"` command to simulate the Insert key during script processing.
If you have trouble with the built in key codes or names, you can alternately create a keyboard profile (Options menu) and map specific byte sequences to the keys, e.g. if the predefined keys from the table below do not suit you, you could remap F-keys for a VT220 or Xterm application with values from the following internet page: Common F-Key Mappings.

1.11.10 ZOC-REXX Commands/Functions

Overview
The section below lists the ZOC extensions to the REXX scripting language in alphabetical order (alternately there is also a list of ZOC-REXX Commands by Category. These commands extend the native REXX programming language. REXX in itself is a full featured programming language, offering the usual constructs like variables, decisions, loops, etc.
To help discern between native REXX and the ZOC-Extensions, the examples below display the native elements of the REXX language in all-uppercase, e.g. `CALL`, `IF`, `THEN`, `SAY`. You will find an overview of such commands and functions in REXX Language Elements.
The ZOC-commands extend the REXX language and offer terminal emulation specific functions. Their names start with with *`Zoc`* and below these commands are written in CamelCase, e.g. `ZocConnect`, `ZocSend`, etc. User defined names (variables) are expressed all lowercase, e.g. `thename= ZocAsk("What is your name");`

Invoking ZOC-REXX Extensions
Generally there are two types of ZOC-commands: Those which return a value or result (e.g. `ZocAsk`, these are also called functions) and those which just perform a task but don't provide a return value (e.g. `ZocDelay`).

- ZOC-commands that do not return a value are invoked using the procedure call syntax: *CALL <cmd-name> <arguments>*, e.g. `CALL ZocDelay 2.5`.
- Functions that return values are invoked using the function call syntax: *<result-var>= <cmd-name>(<arguments>)*, e.g. `answer= ZocAsk("Your name?")`

However, if you are not interested in the result value, it is possible to simply discard it by calling functions (commands which return values) using the procedure style invocation. In other words, the statements `CALL ZocDownload "ZMODEM", "SOMEFOLDER"` and `error= ZocDownload("ZMODEM", "SOMEFOLDER")` are both correct. In fact, the ZOC-REXX implementation even allows this variant: `CALL ZocDownload("ZMODEM", "SOMEFOLDER")`

ZOC REXX Extensions

In the list below, the use of brackets in commands titles indicates that you typically need to use the function style (see above), because the return value of the commands is usually significant.

ZocAsk([<title> [, <preset>]])

Show a text input window and read text from user. If the second argument (preset) is provided, the entry field will be preset with this value.

Example:

```
answer= ZocAsk("What is the best terminal?", "ZOC")
IF answer="ZOC" THEN ...
```

See also: ZocDialog, ZocAskPassword, ZocAskFilename, ZocAskFoldername, ZocRequest, ZocRequestList

ZocAskPassword([<title>])

Same as the ZocAsk command, except that it is intended to enter passwords, i.e. the entry field shows typed characters as dots and you cannot preset the field with a default value.

Example:

```
pw= ZocAskPassword("What's your password?")
IF pw=="secret" THEN ...
```

ZocAskFilename(<title> [, <preselected file>])

Display a file selection window and return the filename. If the file dialog is cancelled, the string `##CANCEL##` is returned.

Example:

```
file= ZocAskFilename("Select file to upload", "*.ZIP")
IF file\="##CANCEL##" THEN DO
    CALL ZocUpload "ZMODEM", file
END
```

See also: ZocAsk, ZocDialog, ZocFilename, ZocAskFilenames, ZocAskFoldername, ZocListFiles

ZocAskFilenames(<title> [, <preselected file> [, <delimiter>]])

Display a window that allows selection of multiple files and return the filenames separated by a space character. If the file dialog is cancelled, the string `##CANCEL##` is returned.

The items can then be extracted from the list by using the `ZocString("WORD", idx)` function. If you expect filenames to contain space characters, you need to supply a different delimiter and use the `ZocString("PART", idx, "|")` function instead.

Example:

```
files= ZocAskFilenames("Select file to process", "*.ZIP", "|")

howmany= ZocString("PARTCOUNT", files, "|")
DO i=1 TO howmany
  name= ZocString("PART", files, i, "|")
  SAY i||". NAME= "||name
END
```

See also: ZocFilename, ZocAskFilename, ZocAskFoldername, ZocListFiles, ZocMessageBox, ZocRequest, ZocRequestList

ZocAskFoldername(<title> [, <preselected folder>])

Display a folder selection dialog and return the name of the selected folder. If the dialog is cancelled, the string `##CANCEL##` is returned.

Example:

```
folder= ZocAskFoldername("Select Folder")
IF folder\="##CANCEL##" THEN DO
    SAY folder
END
```

See also: ZocFilename, ZocAskFilename, ZocAskFilenames

ZocBeep [<n>]

Beep n times.

Example:

```
CALL ZocBeep 2
```

ZocClipboard <subcommand> [, <writestring>]

Performs a clipboard function for one of the following subcommands:

READ
Returns the current content of the clipboard

WRITE
Writes the string from the 2nd parameter to the clipboard

Example:

```
clip= ZocClipboard("READ")
newclip= clip||ZocCtrlString("^M^J Zoc was here!")
CALL ZocClipboard "WRITE", newclip)
```

ZocClearScreen
Clears the screen and resets the emulation to its initial state.

Example:

```
CALL ZocClearScreen
```

ZocCommand <subcommand>
Performs a function for one of the following subcommands:

CLS
Clear screen.

CLEARSCROLLBACK
Clear scrollback buffer.

CANCELCONNECT
Cancel a connect request that is currently in progress.

LOADGLOBALCOLORS
Reload the global color table from file (default file name or 2nd Parameter).

SAVEPROGRAMSETTINGS
Permanently stores changes, which were made via the `ZocSetProgramOption` command.

SAVESESSIONPROFILE
Stores changes made via the `ZocSetSessionOption` to the current session profile file (see also `ZocSaveSessionProfile`)

SETMARKEDAREA
Marks an area on screen. After the subcommand provide the word `BLOCK` or `STREAM`, followed by coordinates x1,y1 and x2,y2.

SENDBREAK
Sends a modem break signal (Serial/Modem connections only).

Example:

```
CALL ZocCommand "SAVESESSIONPROFILE"
```

Example:

```
CALL ZocCommand "SETMARKEDAREA", "BLOCK", 0,0, 1,79
```

See also: ZocMenuEvent

ZocConnect [<address>]

Connect to a host via telnet, modem, ssh etc. or read the address to connect to from the user if the parameter is omitted.

The connection method will be the one that is active in the current session profile. Alternately a connection method can be selected in the script via ZocSetDevice.

If the address is an SSH host, you can pass the username and password to the host in the form *CALL ZocConnect "user:pass@ssh.somedomain.com"* and you can also provide the name of a key file after the password separated with a colon, *CALL ZocConnect "user:pass:keyfile@ssh.somedomain.com"*.

Example:

```
CALL ZocSetDevice "Secure Shell"
CALL ZocConnect "harry:alohomora@192.168.1.1:10022"
```

Example:

```
CALL ZocSetDevice "Telnet"
CALL ZocConnect "server.hogwarts.edu"

Call ZocTimeout 20
x= ZocWait("Login:")
IF x=640 THEN SIGNAL waitfailed /* login prompt not received */
CALL ZocSend "harry^M"
x= ZocWait("Password:")
IF x=640 THEN SIGNAL waitfailed /* password prompt not received */
CALL ZocSend "secret^M"

/* At this point we are logged in */

waitfailed:
EXIT
```

See also: ZocConnectHostdirEntry, ZocSetDevice, ZocDisconnect, ZocGetInfo("ONLINE")

ZocConnectHostdirEntry <name>

Makes a connection based on the details of an entry in the ZOC host directory (the host directory entry should not have a Login REXX file assigned to it).

Example:

```
CALL ZocConnectHostdirEntry "Webhost 03"
```

See also: ZocConnect, ZocDisconnect, ZocGetInfo("ONLINE")

ZocCtrlString(<text>)

This function converts a string containing control codes into a string where the control codes are converted into their respective values.

Example:

```
crlf= ZocCtrlString("^M^J") /* results in two byte string hex"0D0A" */
```

See also: ZocCtrlString

ZocDdeClient([<channel>,] <subcommand> [, <parameters>]

This function can be used to interact with other software via DDE (dynamic data exchange). For example MS-Excel supports DDE and the DDE client can be used to retrieve data from Excel worksheets.

Possible <subcommands> are:

INIT

Initializes a DDE connection. INIT is followed by two parameters: DDE server-name and topic. INIT will return either ##ERROR## or a channel number for use with subsequent commands.

EXECUTE

Performs a server-function. EXECUTE is followed by one parameter: execution-command (e.g. an Excel command).

REQUEST

Requests data from the server. REQUEST is followed by one parameter: data-address (e.g. an Excel cell or range).

CLOSE

Closes an existing dde-channel.

Example:

```
chan= ZocDdeClient("INIT", "EXCEL", "Table1")
SAY "INIT: channel= "||chan

IF chan\="##ERROR##" THEN DO
  data= ZocDdeClient(chan, "REQUEST", "R1C1")
  SAY "REQUEST-DATA: "||data
```

```
  CALL ZocDdeClient chan, "CLOSE"
END
EXIT
```

ZocDelay [<sec>]

Wait a given time in seconds or wait 0.2 seconds if the parameter is omitted.

Example:

```
CALL ZocDelay 4.8
```

ZocDeviceControl <string>

This rather arcane command performs an operation that is specific to the current connection type (e.g. to return the signal states of the COM during a Serial/Direct type connection).

Possible control commands for each communication method are described in ZOC Devices.

Example:

```
state= ZocDeviceControl("GETRS232SIGNALS")
```

See also: ZocSetDevice

ZocDialog <subcommand> [, <parameter>]

Performs a dialog related function:

LOAD

Shows a user defined dialog window. The 2nd parameter is a combination of the name of the dialog together with the name of the file containing the dialog template (the file name is either a fully qualified file name or the name of a file located in the same folder as the currently running script file).

SHOW

Shows a user defined dialog window. There can be an optional 2nd parameter as described above for the LOAD command. In this case, the SHOW command will include the LOAD operation.

GET

Returns the value of one of the dialog elements (e.g. the text which the user had typed into an entry field or the state of a checkbox or radiobutton).

SET

Set the value of one of the dialog elements, e.g. the text which the will initially be shown in an entry field or the description of an item. With a checkbox or radiobutton, the values `"##ON##"` or `"##OFF##"` will also change the initial state.

For details about how dialog templates are built, see ZocDialog Templates.

Example:

```
dlgrc= ZocDialog("SHOW", "MAIN@test.dlg")
```

```
IF dlgrc=="##OK##" THEN DO
  name= ZocDialog("GET", "ED1")
  SAY "Hallo "||name
  SAY ZocDialog("GET", "CB1")
  SAY ZocDialog("GET", "P1")
  SAY ZocDialog("GET", "P2")
END
```

Example:

```
dlgrc= ZocDialog("LOAD", "MAIN@test.dlg")
SAY "Dialog load result: "||dlgrc
CALL ZocDialog "SET", "ED1", "foobar"
CALL ZocDialog "SET", "CB1", "##ON##"
CALL ZocDialog "SET", "DD1", "Red"
CALL ZocDialog "SET", "DD2", "Apple|Orange|Grape"
CALL ZocDialog "SET", "DD2", "Orange"
dlgrc= ZocDialog("SHOW")
IF dlgrc=="##OK##" THEN DO
    /&ast; process result &ast;/
END
```

See also: ZocAsk, ZocAskPassword, ZocMessageBox, ZocRequest, ZocRequestList

ZocDisconnect

Disconnects the current connection. Same as `ZocCommand "DISCONNECT"`.

Example:

```
CALL ZocDisconnect
```

See also: ZocConnect, ZocConnectHostdirEntry

ZocDownload(<protocol>[:<options>], <file or dir>)

Download one or more files using a file transfer protocol.

The first parameter is the name of a file transfer protocol (as listed in ZOC's Options→Session Profile→File Transfer dialog).

The exact nature of the second parameter varies depending on the transfer type (see note below).

For a discussion of the protocol options, please see the ZocUpload command further down in this list.

Depending on success or failure, the function returns the string `##OK##` or `##ERROR##`

Example:

```
CALL ZocSetSessionOption "TransferAutoStart=no"
ret= ZocDownload("ZMODEM", "C:\ZOC\INFILES")
IF ret=="##ERROR##" THEN DO
```

```
    CALL ZocBeep 5
    SAY "Download failed."
END
```

Note: The second parameter varies depending on the file transfer type:
XMODEM: Local destination filename.
YMODEM: Local destination folder.
ZMODEM: Local destination folder.
SCP: Remote source file, e.g. `/var/log/somefile.txt`

Note: If you have Auto Transfer enabled in the Options→Session Profile→File Transfer options, and if the remote host starts the transfer before ZOC-REXX processes the ZocDownload command, then two download windows will come up. So you need to make sure you are issuing ZocDownload() before the host starts or make sure that Auto Transfer option is disabled.

Note: If the file has a name that is set for download to the alternate path (in Options→Session Profile→File Handling), the directory parameter is ignored.

See also: ZocUpload

ZocDoString(<commandstring>)

Pass an action code to ZOC for processing.

You can obtain such strings by temporarily mapping something to a key via Options→Keyboard Mapping Profiles and then copying the result from the keyboard profile. This can be done by editing the corresponding keyboard profile file (e.g. Standard.zky), where you will find an action in the form `^XXXXX=....` and which then you can use as argument for the ZocDoString call.

Example:

```
CALL ZocDoString "^EXEC=notepad.exe"
```

See also: ZocMenuEvent, ZocShell, ZocSendEmulationKey

ZocEventSemaphore(<subcommand>[, <signal-id>])

This function can be used to synchronize and exchange signals between multiple REXX scripts (it has no use within a single script). To facilitate this, the function offers a signaling mechanism with 16 signal slots for which other scripts can wait.

Possible <subcommands> are:

RESET

Sets the state of this semaphore to not-fired and clears the signal counter.

FIRE

Sets state to signaled, increases the counter and releases all waiting scripts.

TEST

Returns the number of received signals (i.e. FIRE commands) since the last reset.

WAIT

Wait for a signal. If the semaphore was already fired, the command will return immediately, resetting the signal (see above) returning the number of signals which happened since the last

reset (max 255).
If the semaphore has not yet been signaled (fired), the function will block and wait for a signal or it will return after a timeout (set via ZocZimeout) returning a code of 640 (the behavior of this function is very similar to ZocWait).

<signal-id>
An optional number [1..15] to access/use a different semaphore (default is 0).

Example:

```
Call ZocEventSemaphore "RESET"
/* do some work */
...
/* wait until another script fires the signal */
ret= ZocEventSemaphore("WAIT")
IF ret\=640 THEN DO
   /* signal was received */
END
```

See also: ZocTimeout, ZocWait, ZocGlobalStore

ZocFilename(<command>[, <options>])
This group of commands offers filename operations.

COMBINE <path>[, <path2>], <file>
Combines filename parts to a full filename. If <file> is already a fully qualified name, <path> and optional <path2> are ignored.

EXISTS <filename>
Returns ##YES## or ##NO##, depending on if the file exists.

GETFILE <filename>
Returns the filename part of a full file descriptor.

GETPATH <filename>
Returns the directory part of a full file descriptor.

GETSIZE <filename>
Returns the size of a file.

GETVOLUMELABEL <driveletter>
Returns the drive label for a drive, e.g. "C:" (Windows only).

ISFOLDER <pathname>
Returns ##YES## or ##NO##, depending on if the given path refers to an existing folder.

RESOLVE <string>
Resolves one of ZOC's special file/path placeholders like %ZOCFILES% or %USERHOME% (the other filename functions do this automatically).

WRITEACCESS <filename>
Returns ##YES## or ##NO##, depending on if a file can be written.

Example:

```
workdir= ZocGetInfo("WORKDIR")
datadir= ZocFilename("RESOLVE", "%ZOCFILES%");

fullfile= ZocAskFilename("Choose File", workdir)
file= ZocFilename("GETFILE", fullfile)
path= ZocFilename("GETPATH", fullfile)

file2= file||".tmp"
target= ZocFilename("COMBINE", path, file2)
IF ZocFilename("EXISTS", target)=="##YES##" THEN DO
    CALL ZocMessageBox "Can't overwrite file "||file2
    EXIT
END
```

ZocFileCopy(<source filename>, <destination>)

Copy a file to a new destination which can either be a file name or folder name.

Wildcards like `*` or `?` are not supported in the source file (see ZocListFiles).

Example:

```
CALL ZocFileCopy "Z:\SALES.DAT", "Z:\SALES.BAK"

ok= ZocFileCopy("C:\DATA\USERFILE.TMP", "C:\BACKUP")
IF ok\="##OK##" THEN EXIT
```

See also: ZocFilename, ZocFileDelete, ZocFileRename, ZocListFiles, ZocShell

ZocFileDelete(<filename>)

Deletes a file. The filename may not contain wildcards (`*` or `?`, see ZocListFiles).

The function returns `##OK##` or `##ERROR##`

Example:

```
ok= ZocFileDelete("C:\DATA\USERFILE.TMP")
IF ok\="##OK##" THEN EXIT

filename= ZocFilename("COMBINE", "%ZOCFILES%", "rexx.log")
CALL ZocFileDelete filename
```

See also: ZocFilename, ZocFileCopy, ZocFileRename, ZocListFiles, ZocShell

ZocFileRename(<oldname>, <newname>)

Renames a file. The renamed file will always remain in the same folder as the original file. Filenames may not contain wildcards (`*` or `?`, see ZocListFiles).

The function returns `##OK##` or `##ERROR##`

Example:

```
ret= ZocFileRename("C:\DATA\USERFILE.TMP", "C:\DATA\USERFILE.TXT")
```

See also: ZocFilename, ZocFileCopy, ZocFileDelete, ZocListFiles, ZocShell

ZocGetHostEntry(<name>,<key>)

Retrieves the key-value pair for a ZOC host directory entry (see ZocSetHostEntry or ZocSetSessionOption commands for more information about such key-value pairs).

Example:

```
pair= ZocGetHostEntry("ZOC Support BBS", "connectto")
PARSE VALUE pair WITH key'="'val'"'
CALL ZocConnect val
```

ZocGetInfo(<what>)

Depending on the parameter, this function can return various bits of information about the current environment and ZOC session.

COMPUTERNAME

The name of the computer on which ZOC is running.

CONNECTEDTO

The name of host directory entry, host name, ip, or phone number to which ZOC is connected.

CONNECTEDTORAW

The actual value of the "connect to" field that was used to initiate the current connection, e.g. `linux.hogwarts.com`.

CONNECTEDTOIP

For connections that are IP based, the actual IP address of the remote host (even if the connection was made by host name).

CURRENTDEVICE

The name of the currently active communication method, e.g. `Telnet`.

CURRENTEMULATION

The name of the currently active emulation, e.g. `Xterm`.

CURRENTLOGFILENAME

Current file name for logging (without path).

CURRENTSCRIPTNAME

Filename and path of the main script which is currently executed (this will not return sub-scripts which are executed via CALL from inside another script).

CURRENTSESSIONPROFILENAME
The filename and full path of the current session profile.

CURSOR-X
The x-position of the cursor (starting with zero).

CURSOR-Y
The y-position of the cursor (starting with zero).

DESKTOPSIZE
The net size of the Windows/macOS desktop in pixels (excluding taskbar, dock, etc.)

DOMAINNAME
The name of this computer' Windows domain or workgroup.

DOWNLOADDIR
The default drive and directory for downloads.

EXEDIR
The drive and directory in which ZOC has been installed.

LASTDOWNLOADEDFILE
The name and path of the last downloaded file.

MARKEDAREA
The start/end position and mode of the marked area in the form *x1,y1,x2,y2,mode* (positions are zero based) or the string `##NONE##`.

ONLINE
Information if ZOC is currently connected to a host: `##YES##`, `##NO##`, `##UNKNOWN##`

OSYS
Returns a string which indicates the operating system and OS version under which ZOC is running, e.g. `Windows 10`.

OWNIP
Returns the computer's IPV4 address in the local LAN or WLAN.

OWNIPS
Returns a list of all IPv4 addresses assigned to the current computer.

PROCESSID
ZOC's system process id.

SCREENHEIGHT
Height (number of lines) of the terminal.

SCREENWIDTH
Width (number of characters) of the terminal.

TRANSFER
Information if a file transfer is currently active: `##YES##`, `##NO##`

TN3270FIELDATTR x y
The TN3270 field attributes and colors of a given screen location (x,y are decimal zero based numbers). The result is in the form *attrbits foreground background attrstrings ...* (numeric values are hexadecimal).

UPLOADDIR
The default drive and directory for uploads.

USERNAME
The login-name of the currently logged in user.

USERID
(same as USERNAME).

VERSION
The current ZOC version, e.g. `7.24`

VERSIONEX
The current ZOC version including beta version (if any), e.g. `7.24b`

WINPOS
A string indicating the position and size of the program window on the screen.

WORKDIR
ZOC's working directory (containing host directory, options, etc).

Example:

```
ver= ZocGetInfo("VERSIONEX")
SAY "ZOC "||ver

CALL ZocTimeout 30
timeout= ZocWait("Password")
IF timeout=640 | ZocGetInfo("ONLINE")<>"##YES##" THEN DO
    SIGNAL PANIC /* disconnected! */
END
```

ZocGetProgramOption(<key>)
Retrieves the key-value pair for a ZOC program option (see ZocGetSessionOption and ZocSetProgramOption for more info about key-value pairs).

Example:

```
pair= ZocGetProgramOption("DisconEndProg")
PARSE VALUE pair WITH key"="value
IF value=="yes" THEN DO
    SAY "ZOC will terminate after this session."
END
```

See also: ZocGetSessionOption, ZocSetSessionOption, ZocSetProgramOption

ZocGetScreen(<x>,<y>,<len>) or ZocGetScreen("<alias>")
The ZocGetScreen() function can be used to return characters that are currently displayed in ZOC's terminal window. It returns <len> characters beginning from position <x>,<y> (zero based). If it reaches the right margin, it continues on the next line without adding a CR or LF.

There are also a few short versions for common combinations of x,y and len:

`ALL`: The whole screen (i.e. from position 0/0 with length SCREENWIDTH*SCREENHEIGHT)
`LEFTOFCURSOR`: The text left of the cursor (position= 0/CURSOR-Y, length= CURSOR-X)
`CURRENTLINE`: The whole line where the cursor is (position= 0/CURSOR-Y, length= SCREENWIDTH)

Example:

```
curline= ZocGetScreen("LEFTOFCURSOR")
SAY "^M"
SAY "The text before the cursor was: "||curline
```

Example:

```
width= ZocGetInfo("SCREENWIDTH")
posy= ZocGetInfo("CURSOR-Y")
screenpart= ZocGetScreen(0,0, posy*width)
IF POS("#", screenpart)=0 THEN DO
    SAY "There is no hash-character on screen above the cursor."
END
```

Example:

```
cx= ZocGetInfo("SCREENWIDTH")
cy= ZocGetInfo("SCREENHEIGHT")

-- loop through all lines on screen
DO y= 0 TO cy-1
    line= ZocGetScreen(0,y, cx)
  -- check each line for some text
    IF POS("**SUCCESS**", line)>=1 THEN DO
        Call ZocMessageBox "Found **SUCCESS* on screen in line "||y
    END
END
```

See also: ZocGetInfo("CURSOR-X"), ZocGetInfo("CURSOR-Y"), ZocGetInfo("SCREENWIDTH"), ZocGetInfo("SCREENHEIGHT")

ZocGetSessionOption(<key>)

Retrieves the key-value pair for a ZOC option from the session profile (see the `ZocSetSessionOption` command for more details).

Example:

```
pair= ZocGetSessionOption("Beep")
```

```
PARSE VALUE pair WITH key"="value
IF value=="no" THEN DO
    SAY "The beep option is turned off."
END
```

Example:

```
pair= ZocGetSessionOption("EnqString")
PARSE VALUE pair WITH key'="'value'"'
SAY "The configured answer to enq is: "||value
```

See also: ZocSetSessionOption, ZocGetProgramOption, ZocSetProgramOption

ZocGlobalStore(<operation>, [<options>])

This group of functions allows permanent storage of values in a file based data pool. This can be used to remember values across various script runs (the operations are atomic and they are protected against concurrent access).

Possible operations are:

SELECT <name>

Select a different global store. With the exception of the name `VOLATILE`, the name will be used to create a file name in the form `<name>Global.ini` in the user data folder or it can point to a different location, e.g. `C:\data\mypool` (which will then also become `mypoolGlobal.ini`).

The name `VOLATILE` refers to a special pool that is only held in memory and which is shared across all sessions. It will lose its values when the last session is closed.

INIT

Clear all values within the data pool.

GET <name>

Returns the value with the given name or an error (see below).

SET <name>, <value>

Stores a value in the storage pool under a given name. Returns `##OK##` or error (see below).

PUT <name>, <value>

Same as SET.

Concurrent access to the store is protected using file locks. This will even work with access from multiple computers if the global store is located on a network drive.

Return codes for the *GET*, *SET* and *PUT* commands include the values `##OK##`, `##LOCKERROR##` if the lock can not be obtained and `##FILEERROR##` if the file can not be accessed.

Example:

```
CALL ZocGlobalStore "SELECT", "MyValStore"
x= ZocGlobalStore("GET", "LASTX")
...
CALL ZocGlobalStore "SET", "LASTX", x

ret=  ZocGlobalStore("SET", "VAL", "OK")
IF ret\="##OK##" THEN SAY "Could not set value"
```

ZocKeyboard(<command> [, <timeout>])

This function allows a REXX script to read keystrokes from the terminal window. It supports the subcommands *LOCK*, *UNLOCK* and *GETNEXTKEY*.

LOCK

Lock the keyboard and prevent user input.

UNLOCK

Unlock the keyboard from a previous *LOCK* state.

GETNEXTKEY

Wait for the next keystroke and return it. You can also specify a timeout in seconds and if successful, GETNEXTKEY will return a string in the form *char|scancode|shift|ctrl|alt*.

`char:` two byte hex number representing the ascii code of the key.
`scancode:` The physical scan code from the keyboard (the scan code can be used to identify functional keys such home, del, f1, f2, etc.).
`shift`, `ctrl` and `alt` are either `0` or `1` indicating if they were held down when the primary key was pressed.

The example below shows how the subcommands are used and how the possible result can be split into its parts and displayed in a user friendly form.

Example:

```
CALL ZocKeyboard "UNLOCK"
ret= ZocKeyboard("GETNEXTKEY", 30)
PARSE VALUE ret WITH hexkey"|"scan"|"shift"|"ctrl"|"alt
key= X2C(hexkey)
SAY "You pressed hex/key: "hexkey"/"key
SAY "Scan code: "scan
SAY "Shift/Ctrl/Alt states: "shift"/"ctrl"/"alt
```

ZocLastLine()

This is a function and returns the last line of text that was received from the point the last ZocWait/ZocWaitMux/ZocWaitLine command was issued to the point when it successfully returned

Example:

```
CALL ZocSend "ATZ^M"
timeout= ZocWaitMux("OK", "ERROR")
IF timeout\=640 & THEN DO
  IF ZocLastLine()\="OK" THEN SIGNAL error
    CALL ZocConnect "555 3456"
END
```

Note: In many cases, using ZocReceiveBuf instead will be the more flexible choice. ZocGetScreen("LEFTOFLINE") often also provides a similar result.

Note: Also see the examples in the description for ZocWaitLine

ZocListFiles(<path\mask> [, <separator>])

The ZocListFiles function will retrieve a list of filenames for a directory.

The first parameter is a directory name and wildcard mask (e.g. `"c:\data\*.*"`).

The function will return a string which contains the number of files and the file names separated by space characters, e.g. `"3 download.zip sales.txt foobar.fil"`

This allows easy access to the parts of the string via REXX's WORD function (see example below). If you expect filenames to contain space characters you can provide a different list separator as the second parameter. E.g. a separator of "|" will return the string `"3 download.zip|sales.txt|foobar.pdf"`. In this case you can use `ZocString("PART", purelist, i, "|")` to extract the file names.

Note: The number of filenames returned is limited to 128 and the maximum length of the total string returned is 4096.

Example:

```
files= ZocListFiles("C:\TEMP\*")

howmany= WORD(files, 1)
SAY "Number of Files:" howmany

purelist= SUBSTR(files, LENGTH(howmany)+2)
DO i=1 TO howmany
    SAY "File " i "=" WORD(purelist, i)
END
```

See also: ZocFilename, ZocGetFilename, ZocGetFilenames, ZocGetFolderName, ZocString

ZocLoadKeyboardProfile [<zkyfile>]

Loads and activates a keyboard profile (*.zky file).

Example:

```
CALL ZocLoadKeyboardProfile "Alternate.zky"
```

ZocLoadSessionProfile <optsfile>

Loads and activates a session profile file (*.zoc).

Example:

```
CALL ZocLoadSessionProfile "Zoc4Linux.zoc"
```

ZocLoadTranslationProfile [<ztrfile>]

Loads and activates a character translation profile (*.ztr).

Example:

```
CALL ZocLoadTranslationProfile "7bitgerman.ztr"
```

ZocMath(<function>, <arg>[, <arg2>])

ZocMath calculates the math function based on the argument(s). Valid functions are sin, cos, tan, asin, acos, sqrt, todeg, torad, bitand, bitor, bitxor.

Example:

```
angle = 270
anglerad = ZocMath("torad", angle)
sinresult = ZocMath("sin", anglerad)
lowbits = ZocMath("bitand", 175, 15)
hibits = ZocMath("bitand", X2D(A5), X2D(F0))
```

ZocMenuEvent <menu text> [, <file>]

Perform a function from the ZOC menu. The <menu text> is a text that matches an entry in the ZOC menu. The <file> parameter is an optional file name which some menu events accept rather than prompting the user for a file.

Example:

```
CALL ZocMenuEvent "Paste (no line breaks)"
CALL ZocMenuEvent "Edit REXX Script", "test.zrx"
```

ZocMessageBox(<text> [, <mode>])

Display a message box with the given text (a `^M` in the text creates a line break).

Normally an informational message window with an OK button (mode 0) is shown. Mode 1 shows an error message with an OK button. Mode 2 shows a message with a YES and NO button.

The return value is either `##OK##` or `##YES##` or `##NO##`.

Example:

```
CALL ZocMessageBox "Connect Failed!", 1
ret= ZocMessageBox("The operation failed^M^MTry again?", 2)
IF ret=="##YES##" THEN DO
   ...
END
```

See also: ZocAsk, ZocAskPassword, ZocRequest, ZocRequestList

ZocNotify <text> [, <duration>]

Display a small floating message at the center of the window. If duration is given, controls the time in milliseconds which the window will stay on screen.

Example:

```
CALL ZocNotify "Hello World!", 1500
```

ZocPing(<ip-or-hostname>, <timeout>)

Sends an ICMP (ping) request to the given host and returns when it receives either a reply or when the timeout (in ms) expires. The command either returns the string `##ERROR##` or the roundtrip time in milliseconds.

Example:

```
pingrc= ZocPing("www.emtec.com", 2500)
IF pingrc\="##ERROR##" THEN SAY "Received reply within "||pingrc||" ms"
```

ZocPlaySound <file>

Plays a .WAV file.

Example:

```
CALL ZocPlaySound "ka-ching.wav"
```

ZocReceiveBuf(<buffer size>)

This function makes ZOC collect parts of a session in a memory buffer and returns the previous buffer's contents (if any) as a string.

Initially the buffer has a size of zero, which means that no data is collected. To start data collection you need to call the function with a parameter indicating the size of the next receive buffer. After that, incoming data is added to the buffer until either the buffer is full or until the function is called again. Calling ZocReceiveBuf again will retrieve the buffer's content, reset the content and set a new size for the buffer.

A sequence of calls to ZocReceiveBuf in order to retrieve text from a database will look like this:

Example:

```
/* make a receive buffer of 256 bytes */
CALL ZocTimeout 60

CALL ZocReceiveBuf 256
CALL ZocSend "read abstract^M"
CALL ZocWait "Command>"

/* get the result from the read command and */
/* make a larger buffer to hold the result of */
/* a subsequent command. */
abst= ZocReceiveBuf(4096)
CALL ZocSend "read contents^M"
CALL ZocWait "Command>"

/* get the content and discontinue buffering */
cont= ZocReceiveBuf(0)

/* At this point, both variables (abst and cont) will start
with the word "read" and end with the character ">", containing
whatever data was received between the command and next prompt. */
```

Example:

```
/* read the remote environment variables and extract the TERM= value */

Call ZocReceiveBuf 2048
Call ZocSend "set^M"

/* you will need to wait for the your own prompt here */
Call ZocWait "PROMPT: ~username$"
data= ZocReceiveBuf(0)

/* google for "REXX PARSE COMMAND" to get more details
  on the PARSE command which is used to extract the data */
PARSE VALUE data WITH ."TERM="term .

SAY "The remote term setting is " term
```

Note: If executed via DDE, the ZocReceiveBuf command must be sent as a DdeRequest (rather than DdeExecute)

Note: See also ZocWaitForSeq and ZocString("LINE" ...)

ZocRegistry(<subcommand>[, <options>])

This group of commands allows access to the Windows registry.

OPEN <basekey>, <name>

Return a <hkey> handle for access to a part of the registry. Basekey can be *HKEY_CURRENT_USER* or *HKEY_LOCAL_MACHINE*.

OPEN returns a hkey value which can be used to read/write this part of the registry.

WRITE <hkey>, <value>, <data>

Writes <data> to the part of the registry which is associated with <hkey>. If <data> is provided in the form *"DWORD:n"* the decimal value *n* will be stored as REG_DWORD. If <data> is provided in the form *"BINARX:xxxxxx..."*, then *xxxxxx...* is converted from a hex string to bytes and will be stored as REG_BINARY. Otherwise <data> will be stored as REG_SZ (string).

READ <hkey>, <value>

Read a value from the registry part <hkey>. If the registry value is in REG_DWORD format, the command will return *"DWORD:n"*. If the value is in REG_BINARY format, the command will return *"BINARY:xxxxxx..."*, where *xx...* represents a hex-string (the hex-string can be converted to bytes via the REXX `X2C` function). Values of type REG_SZ will be returned without prefix, i.e. the command will simply return the string from the registry.

ENUM <hkey>, <n>

Returns the <n>th value name from <hkey> or `##ERROR##` if no such value exists.

TEST <hkey>, <value>

This function tests, if the given value exists and returns either `##ERROR##`, or a string in the form `##OK## TYPE nt LENGTH nl`, where nt and nl are a decimal values indicating the type of the entry and the length of the data.

CLOSE <hkey>

Ends access to <hkey>.

Example:

```
hk= ZocRegistry("OPEN", "HKEY_CURRENT_USER", "Software\Emtec\ZOC9")
IF hk=="##ERROR##" THEN EXIT
CALL ZocRegistry "WRITE", hk, "Test01", "Hello World"
CALL ZocRegistry "WRITE", hk, "Test02", "DWORD:1"
CALL ZocRegistry "WRITE", hk, "Test03", "BINARY:5A4F43"
SAY ZocRegistry("TEST", hk, "%ZOC%")
homepath= ZocRegistry("READ", hk, "%ZOC%")
SAY "ZOC installed in "||homepath
i= 0
DO  FOREVER
  x= ZocRegistry("ENUM", hk, i)
  IF x=="##ERROR##" THEN LEAVE
  i= i+1
  SAY "Value named "||x
END

CALL ZocRegistry "CLOSE", hk
EXIT
```

ZocRequest(<title>, <opt1> [, <opt2> [, <opt3>]])

Displays a dialog window with options and returns a string containing the selected option.

Example:

```
answer= ZocRequest("What do you want?", "Milk", "Honey")
```

```
IF answer=="Milk" THEN DO
    ...
END
```

See also: ZocAsk, ZocAskPassword, ZocMessageBox, ZocRequestList

ZocRequestList(<title>, <opt1> [, ...]])

Displays a dialog window with a list of options and returns the index of the selected option (or -1 for Cancel). If only one option is passed to the function, it is considered as a list of choices separated by vertical bars.

Example:

```
answer= ZocRequestList("Please select!", "Beer", "Wine", "Whiskey", "Gin")
IF answer=3 THEN DO
    ...
END

answer= ZocRequestList("Please select!", "Beer|Wine|Whiskey|Gin")
IF answer=3 THEN DO
    ...
END
```

See also: ZocAsk, ZocAskPassword, ZocMessageBox, ZocRequest

ZocRespond <text1> [, <text2>]

Send *text2* whenever *text2* is received while REXX is processing a ZocDelay or ZocWait command.

A maximum of 64 Respond commands can be active simultaneously. <text1> must not contain carriage returns or line feeds.

If *text2* is omitted or empty the response command for <text1> is cleared. If *text1* is empty ("") all responses are cleared.

Example:

```
/* Wait for 'Command' and auto-skip all possibly prompts in between */
CALL ZocRespond "Enter", "^M"
CALL ZocRespond "More", "^M"
timeout= ZocWait("Command")
/* Clear responders */
CALL ZocRespond "Enter"
CALL ZocRespond "More"
```

The above example waits until the text `Command` is received. While waiting, all `Enter` and `More` prompts are answered automatically by sending Enter. After the Wait is satisfied, the respond commands are cancelled.

ZocSaveSessionProfile [<optsfile>]

Save the current session profile to file. If the <optsfile> parameter is omitted, ZOC will ask the user for a filename.

Example:

```
CALL ZocSetSessionOption "JumpScroll=3"
CALL ZocSaveSessionProfile "Fastscroll.zoc"
```

See also: ZocCommand("SAVESESSIONPROFILE")

ZocSend <text>

Sends the given text to the remote host.

Internally the text sending is processed as a series keystrokes rather than sending it directly through the low level communication channel. The send speed is based on the text sending option int Options→Session Profile→Text Sending. If you need a faster, more direct version of this command, please use ZocSendRaw

Also, if the text contains control codes (e.g. ^M for Enter), are replaced with their respective values. For nearly all emulations these control codes are based on the control codes. Exceptions are the TN3270 and TN5250 emulations, where ^M is interpreted as Newline/FieldExit, ^I as Tab and ^Z as Transmit/Enter.

Example:

```
/* send JOE USER<enter>*/
CALL ZocSend "JOE USER^M"
```

Example:

```
/* Unix login Sequence */
CALL ZocWait "login:"
CALL ZocSend "harry^M"
CALL ZocWait "password:"
CALL ZocSend "alohomora^M"
```

Example:

```
/* 3270/5250 example */
CALL ZocSetCursorPos 12,5
CALL ZocSend "Freddie"
CALL ZocSendEmulationKey "NewLine"
CALL ZocSend "Elm Street"
CALL ZocSendEmulationKey "Enter"

/* 3270/5250 same as above */
```

```
CALL ZocSend "Freddie^MElm Street^Z"
```

See also: ZocSendRaw, ZocSendEmulationKey.

ZocSendEmulationKey <keyname>

Send the code that represents a special key in the current terminal emulation, e.g. send F17 from a VT220 emulation or Attn under TN3270.

The key names are described in the Key Names Appendix.

Example:

```
CALL ZocSendEmulationKey "f17" /* Send F17 based on current emulation */
```

ZocSendRaw <datastring>

This command sends the data from datastring in untranslated form. The command does not translate control sequences like ^M. If you need to send such codes, you will have to use the REXX string functions like X2C(<hexcode>) or ZocCtrlString to create a corresponding character values (e.g. X2C(0D) for Enter).

ZocSendRaw may be useful if you want to send binary data to a host. For example, if you want to send 42 01 00 05 41 43 (hex) through the communication channel, you can do this as in the 2nd part of the example below.

Example:

```
CALL ZocSendRaw "Login"||X2C(0d) /* Login<enter> */

/* Four times the same result: */
CALL ZocSendRaw X2C(420100054143)
CALL ZocSendRaw "B"||X2C(01)||X2C(00)||X2C(05)||"AC"
CALL ZocSendRaw ZocCtrlString("B^A^@^EAC")
CALL ZocSend "B^A^@^EAC"
```

See also: ZocSend, ZocSendEmulationKey, ZocCtrlString

ZocSessionTab(<subcommand>, <parameters>)

This function allows a REXX script to access or manipulate session tabs. The subcommand defines the action, the parameters depend on the subcommand.

CLOSEATEXIT

Close the current session tab when the script exits.

Example: `CALL ZocSessionTab "CLOSEATEXIT"`

CLOSETAB

Close the session tab with the given index (zero for the leftmost tab or -1 for the current tab).

Example: `CALL ZocSessionTab "CLOSETAB", 2`

GETCOUNT

Returns the number of session tabs.

Example: `howmany= ZocSessionTab("GETCOUNT")`

GETCURRENTINDEX

Returns the index of the tab in which this script is running.

Example: `myidx= ZocSessionTab("GETCURRENTINDEX")`

GETINDEXBYNAME, <name>

Returns the index of the first tab that has a given title or -1 if none was found.

Example: `srvidx= ZocSessionTab("GETINDEXBYNAME", "My Server")`

GETNAME, <index>

Return the name of the tab with a given index (zero for the leftmost tab) from the 1st parameter. An index of -1 refers to the session in which the script is running.

Example: `name= ZocSessionTab("GETNAME", -1)`

ISCONNECTED, <index>

Returns `##YES##` or `##NO##` depending on if the tab with the given index has an active connection.

Example: `yesno= ZocSessionTab("ISCONNECTED", 2)`

NEWSESSION, <title>, <activate>, <sessionprofile>[, <connectto>, <script>]

Create a new session and return the new session's index.

Parameters:
<title>: A string naming the tab.
<activate>: 0 or 1 depending on if the new tab should be in the background or active.
<sessionprofile>: name of a session profile (e.g. `MyProfile.zoc`) or `##NULL##` for default.
<connectto>: A string as described with the /CONNECT command line parameter or a string prefixed with `CALL:` followed by the name of an entry in the host directory or `##NULL##` for none.
<script>: The name of a script file to run in the new tab.

Example:

```
idx1= ZocSessionTab("NEWSESSION", "Test1", 1, "##NULL##", "CALL:My
Server")

idx2= ZocSessionTab("NEWSESSION", "Test2", 0, "SSHProfile.zoc", "SSH!
Harry:alohomora@ssh.hogwarts.edu")
```

```
idx3= ZocSessionTab("NEWSESSION", "Test3", 1, "##NULL##", "##NULL##",
"test.zrx")
CALL ZocSessionTab "SETCOLOR", idx3, 5

idx4= ZocSessionTab("NEWSESSION", "Test4", 1, "Standard.zoc", "TELNET!
smtp.hogwarts.edu:25", "test.zrx")
```

MENUEVENT, <index>, <menu>

Performs a command from the ZOC menu in a different session. The <index> parameter indicates the session (see the description of the <index> parameter for GETNAME), the <menu> parameter is the same as for `ZocMenuEvent`.

Example: `CALL ZocSessionTab "MENUEVENT", idx, "Disconnect"`

RUNSCRIPT, <index>, <title>

Starts a script in a different session. The <index> parameter indicates the session (see description of the <index> parameter for GETNAME). Please note that this will not work for sessions which already have a script running (including the session/script which issues the ZocSessionTab("RUNSCRIPT", ...) command.

Example: `CALL ZocSessionTab "RUNSCRIPT", newidx, "configure.zrx"`

SEND, <index>, <text>

Sends text to the session with the given index (similar to using the ZocSend command). The <index> parameter indicates the session (for details see description of the <index> parameter for GETNAME).
Example: `CALL ZocSessionTab "SEND", 2, "exit^M"`

SETCOLOR, <index>, <color>

Sets the color of the tab with a given index. The <index> parameter indicates the session (for details see the description of the <index> parameter for GETNAME). The color is a number between 0 and 7.

Example: `CALL ZocSessionTab "SETCOLOR", -1, 4`

SETBLINKING, <index>, <blinkflag>

Activates or deactivates the blinking for the tab with a given index. The <index> parameter indicates the session (for details see the description of the <index> parameter for GETNAME). The <blinkflag> is either 1 or 0.

Example: `CALL ZocSessionTab "SETBLINKING", -1, 1`

SETNAME, <index>, <title>

Sets the name of the tab with a given index. The <index> parameter indicates the session (for details see the description of the <index> parameter for GETNAME).

Example: `CALL ZocSessionTab "SETNAME", -1, "This Session"`

SWITCHTO, <index>

Activate the tab with a given index. The <index> parameter indicates the session (for details see the description of the <index> parameter for GETNAME).

Example: `CALL ZocSessionTab "SWITCHTO", 2`

WRITE, <index>, <text>

Write text to the screen of the session with the given index (similar to using the ZocWrite command within that tab).
The <index> parameter indicates the session (for details see description of the <index> parameter for GETNAME).
Example: `CALL ZocSessionTab "WRITE", 1, "Progress 10%^M"`

Example:

```
/* ZocSessionTab sample: send a text to all tabs */

text= ZocAsk("Command to send to all other tabs:")
IF text\="##CANCEL##" THEN DO
  n= ZocSessionTabs("GETCOUNT")
  c= ZocSessionTabs("GETCURRENTINDEX")
  SAY n
  DO i=0 TO n-1
  IF i==c THEN ITERATE
    name= ZocSessionTabs("GETNAME", i)
    CALL ZocSessionTabs "SEND", i, text||"^M"
    SAY "Sent "||text||" to "||name
  END
END
```

ZocSetAuditLogname <filename>

Set the file name for the audit log (a logfile that can not be turned off by the user, also see the setting in the ADMIN.INI file in the program folder) or turn off audit logging with "" (empty string) as a filename.

ZocSetAutoAccept 1|0

For those connection types that support it (e.g. Telnet), make the communication method accept incoming connections.

Example:

```
CALL ZocSetDevice "Telnet"
CALL ZocSetAutoAccept 1 /* accept calls */
```

ZocSetCursorPos <row>, <column>

This command moves the cursor to the given position on a TN3270 screen (in other emulations the command is ignored). The positions are 1-based, i.e. the top/left position on screen is 1/1.

Example:

```
CALL ZocSetCursorPos 1,15
Call ZocSendEmulationKey "Enter"
```

ZocSetDevice <name> [, <commparm-string>]

Change the connection type (communication device). The name must be one of the names from the list in Options→Session Profile→Connection Type.

The optional commparm-string contains options, which can further tweak the operation of that connection (e.g. setting Telnet-specific options, see ZocSetDeviceOpts for information about how to obtain a commparm-string). If the comm parameters are omitted, ZOC uses the options which are set for the connection type in the currently active session profile.

Example:

```
CALL ZocSetDevice "Telnet"
CALL ZocConnect "bbs.channel1.com"
```

Example:

```
CALL ZocSetDevice "SERIAL/MODEM", "[1]COM3:57600-8N1|9|350"
```

ZocSetDeviceOpts <parameter-string>

This is a rather arcane command, which sets the options for a communication method (device) directly from REXX. However, since the options strings for the device are not standardized, in order to find a specific parameter string, you will need to set the options manually in the session profile dialog and then query the current connection type's parameter string by pressing Shift+Ctrl+F10 in ZOC's main window.

Let us assume you want to start a modem session on COM3 with 57600 bps, RTS/CTS and Valid-CD active and a break time of 350ms.

1. Go to Options→Session Profile→Connection Type, select Serial/Modem and the set these options.
2. Close the session profile window using 'Save'
3. Press Shift+Ctrl+F10
4. The status output from that connection type will show the current device-parameter [1]COM3:57600-8N1|9|350 which you can use as a parameter to the ZocSetDeviceOpts command.

Example:

```
/* Set serial options to:
   COM3, 57600-8N1, RTS/CTS, Valid-CD, 350ms */
CALL ZocSetDeviceOpts "[1]COM3:57600-8N1|9|350"
```

Example:

```
/* Select telnet and set options for
   "Start session with local echo" */
CALL ZocSetDevice "TELNET"
CALL ZocSetDeviceOpts "[3]12"
```

ZocSetEmulation <emulationname>[, <emuparm-string>]

The ZocSetEmulation command allows a script to activate a different terminal emulation. The parameter can be one of the names shown in the Emulation section of the session profile dialog, e.g. `VT220`, `TN3270`, etc.
The optional emuparm parameter contains optional settings that configure emulation dependent options (otherwise the settings from the current session profile are used).

Example:

```
CALL ZocSetEmulation "Xterm"
```

ZocSetHostEntry "name", "<key>=<value>"

Sets a value for a host directory entry from a key-value pair. To find actual names for these key-value pairs, please check the file HostDirectory.zhd (stored in the ZOC data folder) using an editor. The file contains all the key-value pairs that make up your host directory.
If instead of a key-value pair, you pass the string "##NEW##", a new host directory entry with that name will be created (if it does not yet exist). You can then configure it with subsequent calls that provide key-value pairs.
See the `ZocSetSessionOption` command for more background about key-value pair handling in general.

Example:

```
CALL ZocSetHostEntry "MyBBS", "emulation=1"

pair= ZocGetHostEntry("ZOC Support BBS", "calls")
PARSE VALUE pair WITH key"="value
value= value+1
CALL ZocSetHostEntry "ZOC Support BBS", "calls="||value

/* mind the mixture of quotes in the commands below */
value= "555-1234"
CALL ZocSetHostEntry "ZOC Support BBS", 'connectto="555-1234"'
CALL ZocSetHostEntry "ZOC Support BBS", 'connectto="'||value||'"'
```

Example:

```
name= "My Router"
```

```
Call ZocSetHostEntry name, "##NEW##"
Call ZocSetHostEntry name, 'connectto="192.168.1.1"'
Call ZocSetHostEntry name, 'username="root"'
Call ZocSetHostEntry name, "deviceid=9"
Call ZocSetHostEntry name, "emulationid=3"
```

See also: ZocSetSessionOption, ZocGetHostEntry

ZocSetLogfileName <filename>

Set new name for logging.

Example:

```
CALL ZocSetLogfileName "Today.LOG"
```

ZocSetLogging 0|1 [,1]

Suspend/resume logging. If a second parameter with value `1` is given, ZocSetLogging will suppress the notification window.

Example:

```
CALL ZocSetLogging 1
```

ZocSetMode <key>, <value>

This commands allows you to set operation modes for some of the script commands. The following modes are supported:

SAY

A value of *RAW* sets the SAY command to *not* convert control codes like `^M` into their respective character values. A value of *COOKED* switches back to standard behavior.

Example:

```
CALL ZocSetMode "SAY", "RAW"
```

RESPOND

A value of *RAW* sets the respond command to *not* convert control codes like `^M` into their respective values for both sides of the command.

Example:

```
CALL ZocSetMode "RESPOND", "RAW"
```

ZocSetProgramOption "<key>=<value>"

This command modifies a ZOC option from the Options→Program Settings dialog. (similarly ZocSetSessionOption modifies the Options→Session Profile). The function basically works the same as ZocSetSessionOption but the underlying file which contains the key-value pairs is Standard.zfg. You can load the file into an editor and you will see that the key names are mostly self

explanatory. If you have problems finding a specific key or value, you can start the REXX script recorder, change the options, stop the recording and look at the resulting script.

Example:

```
CALL ZocSetProgramOption "SafAskClrCapt=yes"
CALL ZocSetProgramOption 'ScriptPath="ZocREXX"'  /* mind the quotes */
CALL ZocSetProgramOption 'ScriptPath="'||pathvar||'"'  /* mind the quotes
*/
```

See also: ZocCommand("SAVEPROGRAMSETTINGS"), ZocGetSessionOption, ZocSetSessionOption, ZocGetProgramOption

ZocSetScriptOutputDestination "DEFAULT|DATASTREAMBROWSER|FILE:<filename>"

This command allows you to redirect the output of a REXX script to the datastream browser (View-menu) or to a file. This affects output from the REXX commands SAY and TRACE. Output from ZocWrite will still appear in the terminal area.

Example:

```
CALL ZocSetScriptOutputDestination "DATASTREAMBROWSER"
```

Siehe auch: ZocWrite, ZocWriteln

ZocSetSessionOption "<key>=<value>"

Sets a ZOC option from the Options→Session Profiles window, based on a key-value pair. To find out more about the key-value pairs, please have a look at the contents of the Standard.zoc file (or any other *.zoc session profile). The file contains all the key-value pairs, which make up a session profile.

If you are not sure what the value of a certain key means, just set the option you want to change in the options window, then click Save and check the options file for the new key-value pair. Or you can also start the REXX script recorder, then make the changes to the session profile and then stop recording and look at the resulting script.

Example:

```
CALL ZocSetSessionOption "JumpScroll=3"
CALL ZocSetSessionOption "ShowChat=no"
CALL ZocSetSessionOption 'MdmIni="ATZ^M"'  /* mind the quotes */
CALL ZocSetSessionOption 'TransAutoRemove="'||valvar||'"'  /* mind the
quotes */
```

Note: ZocSetSessionOption/ZocGetSessionOption will only work for options from the Options→Session Profile dialog. To change Options→Program Settings, use ZocSetProgramOption.

See also: ZocCommand("SAVESESSIONPROFILE"), ZocSaveSessionProfile, ZocGetSessionOption, ZocGetProgramOption, ZocSetProgramOption

ZocSetTimer <hh:mm:ss>

Set connection timer to given time. If called with an empty parameter, the function will return the current duration of the session timer in seconds. Calling the function with a parameter string `"STOP"` will hold the timer and `"RESUME"` will continue with a held timer.

Example:

```
CALL ZocSetTimer "00:00:20"
```

ZocSetUnattended 0|1

Enable or disable ZOC's unattended mode (same as command line parameter /U).

Example:

```
CALL ZocSetUnattended 1
```

ZocShell <command>, [<viewmode>]

Execute a command or program in a shell window via cmd.exe /c <command> (Windows) or /bin/bash -c "<command>" (macOS). This is similar to using to REXX's native `ADDRESS CMD "<command>"` and essentially allows the execution of anything that you can execute in the shell/terminal window of the operating system.

Windows only: The optional viewmode parameter controls how the black shell window is shown:
0= normal, 1= hidden, 2= minimized, 3= maximized.

In many cases ZocShell is identical to using the native REXX shell interface via `ADDRESS CMD "<command>"`

Example:

```
CALL ZocShell "DEL FILE.TMP"
CALL ZocShell "touch /tmp/file.lck", 1
```

See also: `ZocShellExec`, `ZocShellOpen`, `ZocFileDelete`, `ZocFileRename`

ZocShellExec <command>[, <viewmode>]

Execute a program directly (i.e. without passing it through the shell's command interpreter, thus avoiding the overhead of running the shell). Under Windows this only works for .com and .exe files (i.e. it will not work for .CMD scripts and internal shell commands like DEL, REN etc.).

The optional viewmode parameter controls how the application window is shown:
0= normal, 1= hidden, 2= minimized, 3= maximized.

ZocShellExec also returns the exit code of the application to the REXX script.

Example:

```
CALL ZocShellExec 'notepad.exe "somefile.txt"'
```

See also: ZocShell, ZocShellOpen

ZocShellOpen <filename>

This function is somewhat equivalent of a double click on a file, because it passes a filename to the operating system with a request to open it. The operating system will then start the program which is associated with the type of that file (e.g. a PDF viewer for PDF files or Notepad for TXT files).

Alternately, an URL can be passed as a parameter instead of a filename.

Example:

```
CALL ZocShellOpen 'C:\DOWNLOADS\Report.pdf'
```

Example:

```
CALL ZocShellOpen 'https://www.emtec.com/'
```

See also: ZocShell, ZocShellExec

ZocString(<subcommand>, <inputstring>, <p1> [, <p2>])

This is a function to manipulate a string and return a modified copy. The subcommand and the parameters <p1> and <p2> control the modification.

LINE

Return the <p1>th element of <inputstring> which is delimited by a line feed (hex 0A) and stripped of leading or trailing carriage return characters (hex 0D) (simply speaking, this refers to the <p1>th line). This is useful to parse multiline results from a ZocReceiveBuf call, e.g. `name= ZocString("LINE", data, 4)` will return the 4th line from the data variable.

LINECOUNT

Returns the number of elements in <inputstring> which are accessible as LINE.

LOAD

Returns the content of the file with name <inputstring> (the file is loaded in text mode, line endings are converted to LF (hex 0A)). The LINE subcommand of ZocString can be used to extract lines from the result string.

SAVE

Saves the content of string <p1> in a file with name given in <inputstring> (the file is saved in text mode, LF line endings are converted to CR/LF under Windows).

MIME-ENCODE

Converts <inputstring> to base-64/MIME.

MIME-DECODE

Converts <inputstring> form base-64/MIME.

UTF8-ENCODE

Converts <inputstring> from an 8-bit string to UTF8.

UTF8-DECODE

Converts <inputstring> form an UTF8-string to 8-bit.

AES256-ENCRYPT

Encrypts the string <p1> via AES256 using the encryption-key from <inputstring>. The result is MIME-encoded.

AES256-DECRYPT
Decrypts the MIME-encoded string from <p1> using the encryption-key <inputstring>.

PART
Return the <p1>th part of <inputstring> which is delimited by <p2>, e.g. `name= ZocString("PART", "Anne|Charly|Joe", 2, "|")` will return "Charly".

PARTCOUNT
Return the number of <p1>-delimited parts in <inputstring>, e.g. `count= ZocString("PARTCOUNT", "Anne|Charly|Joe", "|")` will return 3.

REPLACE
Return a copy of <inputstring> where all occurrences of <p1> are replaced by <p2>, e.g. `betterstr= ZocString("REPLACE", str, "HyperTerminal", "ZOC")`

REMOVE
Return a copy of <inputstring> where all occurrences of <p1> are removed, e.g. `betterstr= ZocString("REMOVE", str, "HyperTerminal")`

REMOVECHARS
Return a copy of <inputstring> from which all characters of <p1> were removed, e.g. `str= ZocString("REMOVECHARS", str, "0123456789"||X2C(09))` removes all digits and tab characters from STR.

TAB
Return the <p1>th element of <inputstring> which is delimited by a tab-character, e.g. `name= ZocString("TAB", tabbed_data, 2)`

TABCOUNT
Return the number of elements of <inputstring> accessible as TAB.

WORD
Return the <p1>th element of <inputstring> which is delimited by a space, e.g. `name= ZocString("WORD", "The quick brown fox", 3)` will return "brown". Same as REXX's native WORD() function.

WORDCOUNT
Return the number of elements of <inputstring> accessible as WORD.

Example:

```
CALL ZocReceiveBuf 1024
CALL ZocSend "ps^M"
CALL ZocWait "$"
data= ZocReceiveBuf(0)

/* display result list line by
   line but ignoring the first*/
howmany= ZocString("LINECOUNT", data)
DO i=2 TO howmany
  SAY ZocString("LINE", data, i)
END
```

Example:

```
key= "Secret.740.$%&"
n= ZocString("AES256-ENCRYPT", key, "Hello World!")
SAY "Encoded: "||n
n2= ZocString("AES256-DECRYPT", key, n)
SAY "Decoded: "||n2
```

See also: ZocCtrlString

ZocSuppressOutput 0|1

Enables or disables suppressing of screen output. This command allows you to send/receive characters without any screen activity. Logging to the capture buffer and to file is also suppressed.

Output suppression is automatically reset to normal when the script ends or upon disconnecting from a host.

ZocSynctime <time>

This command lets you define the sync-time period (the default time is 250ms).

Background: Because REXX programs are running parallel to ZOC, the main window may receive and process more incoming data while REXX is performing its own work. This means that in some situations it is possible that text, which you expect to be received and which are going to wait for, has actually already scrolled by.

A typical example for this is a loop that attempts to process all incoming lines of text.

Example:

```
DO FOREVER
    timeout= ZocWaitLine()
    IF timeout\=640 THEN DO
        data= ZocLastLine()
        /* process data in some way */
    END
END
```

In this example ZOC could receive more text while the REXX program still processes the data from ZocLastLine and before it is ready to loop and issue the next ZocWaitLine.

To address this problem, ZOC's processing of incoming traffic is suspended for a short time period (the sync-time) whenever a Wait-command has been satisfied, thus giving the REXX program time to process the result and to get ready to wait for more data.

After a Wait-command (ZocWait, ZocWaitMux, etc.), ZOC will resume processing of incoming data either if the sync-time has elapsed or if the REXX program issues another Wait-command or if another command is processed, which needs to interact with the main window (ZocWrite, ZocSend,

etc. but also SAY, TRACE, because those output to the main window).

Important: Instead of increasing the sync-time, in loops like the above, you should consider to merely collect and store the data for later processing (the ZocWaitLine command has an example of how to do this properly). An even better alternative is the use ZocReceiveBuf to catch all the data in one packet.

See also: ZocWait, ZocWaitIdle, ZocWaitLine, ZocWaitMux, ZocReceiveBuf, ZocLastLine

ZocTerminate [<return-code>]

Causes ZOC to end the program and close after the REXX program has ended. Usually an `EXIT` command will follow ZocTerminate.

If you supply the return-code parameter, the ZOC process will return this value to the operating system or to the calling program.

Example:

```
CALL ZocTerminate
EXIT
```

ZocTimeout <sec>

Set maximum time to wait for a ZocWait/ZocWaitMux/ZocWaitLine/ZocEventSemaphore command.

Example:

```
/* Make subsequent ZocWait commands expire after 30 seconds */
CALL ZocTimeout 30

/* Wait until the host sends 'ready' of until the timeout expires */
timeouttimeout= ZocWait("ready")
IF timeout=640 THEN SAY "ready-prompt not received within 30 seconds"
```

See also: ZocWait, ZocWaitIdle, ZocWaitLine, ZocWaitMux, ZocEventSemaphore, ZocSyncTime

ZocUpload <protocol>[:<options>], <path/filename>

Start to upload a file using the given file transfer protocol.

If the filename contains no path, it is taken from the standard upload folder, otherwise, if the path is relative, it is accessed relative to ZOC's program folder or relative to the upload folder.

For file transfer protocols which support multiple files (Ymodem, Zmodem), the file parameter may contain wildcards. Multiple filenames may be provided as one string in which the file names are separated by vertical bars `*.pdf|somefile.txt`

The protocol name is `ASCII`, `BINARY` or the name of the file transfer protocol from the

Options→Session Profile→File Transfer dialog, e.g. `Zmodem` or `Kermit`.

Depending on the success of the transfer, ZocUpload will return `##OK##` or `##ERROR##`.

Example:

```
CALL ZocUpload "ZMODEM", "id_dsa.pub"
```
uploads id_dsa.pub from the local upload folder to the remote host using the Zmodem protocol.

Example:

```
success= ZocUpload("XMODEM", "updates.zip")
```
uploads the file `updates.zip` from the upload folder via Xmodem protocol and obtains a success indicator (`##OK##` or `##ERROR##`).

Example:

```
CALL ZocUpload "BINARY", "CNC-CONTROL.DAT"
```
sends the contents of the file directly without any translation or protocol.

Example:

```
CALL ZocUpload "ASCII", "commands.txt"
```
uploads commands.txt using the text sending options which are currently configured in the session profile.

Example:

```
CALL ZocUpload "ASCII:CRONLY+10", "\FAR\AWAY\LIST.TXT"
```
uploads LIST.TXT via text/ascii transfer using CR-only as the end of line marker and a character delay of 10ms.

Example:

```
CALL ZocUpload "ASCII:1+3", "HERE\SOME.DATA"
```
uploads the file SOME.DATA via ascii transfer with CR/LF translation and a character delay of 3ms.

Transfer Options
For all protocols (except IND$FILE, ASCII and BINARY, see below), you can set the optional options by copying a string from a session profile which configures that protocol. To do this, set your desired protocol options in ZOC's Options→Session Profile→File Transfer dialog and query the current parameter string by pressing Shift+Ctrl+F10 in ZOC's main window.

Example: If you want to perform a file transfer via Xmodem using the CRC option and 1KB data blocks, you
1. Go to ZOC's Options→Session Profile→Transfer dialog, select Xmodem and configure these options.
2. Close the session profile window (click 'Save')
3. In ZOC's main window press Shift+Ctrl+F10
4. The status output will show the current transfer options "[0]kc" which you can use as a parameter to the ZocUpload command.
5. The REXX command will be `CALL ZocUpload "XMODEM:[0]kc", "datafile.zip"` (please note that the options are case sensitive).

Transfer Options ASCII
For the file transfer in ASCII mode (equivalent of Transfer→Send Text File), the options parameter has the format "mode+chardelay+linedelay", where mode specifies the end of line translation as a number or text (ASIS= 0, CRLF= 1, CRONLY= 2, LFONLY= 3) and delays are the per character and per line send delays, e.g. valid parameters would be `2+5+100` or `CRONLY+5+100` (which means CR only with 5 milliseconds per character and 100 ms extra delay at the end of each line).

Transfer Options BINARY
For a file transfer in BINARY-mode (equivalent of Transfer→Send Text File), the options parameter can be a number. This number sets the character delay in milliseconds (otherwise the text sending delay from the session profile will be used).

Transfer Options IND$FILE
The IND$FILE file transfer type is an exception to the above. The options part for IND$FILE actually contains the corresponding host command to perform the transfer (you can look up this command at the bottom of the dialog which is shown when performing a manual file transfer via IND$FILE), e.g.:

```
CALL ZocUpload "IND$FILE:TSO IND$FILE PUT 'userid.projects.asm(report)'
ASCII CRLF", "report.asm"
```

ZocWait(<text>)

Waits until the given text is received. If ZocWait times out (see `ZocTimeout`), it returns a value of 640.

Example:

```
CALL ZocTimeout 20
timeout= ZocWait("Password")
IF timeout=640 THEN SAY "Password prompt not received within 20 seconds"
ELSE CALL ZocSend "alohomora^m"
```

Example:

```
CALL ZocTimeout 10

timeout= ZocWait("enter command>")
IF timeout=640 THEN SIGNAL the_end
CALL ZocSend "ENABLE FIREWALL^M"

timeout= ZocWait("enter command>")
IF timeout=640 THEN SIGNAL the_end
CALL ZocSend "ENABLE IPFILTER^M"

timeout= ZocWait("enter command>")
IF timeout=640 THEN SIGNAL the_end

SAY "Firewall and Ip-Filter activated!"

the_end:
Call ZocDisconnct
```

Note: ZOC automatically filters ANSI/VTxxx/etc. control sequences from the data stream to avoid interference with the ZocWait command (see ZocWaitForSeq).

Note: If you are using ZocWait from a DDE controller, the command must be sent as via DDE-Request rather than via DDE-Execute.

Note: With the TN3270 and TN5250 emulations, the command can be used to wait for `"^Z"`, which will be used as a signal that indicates that the emulation is ready to accept input again.
, because these emulations are transmitting whole screens at a time, you should only issue one ZocWait per screen. In other words, if a ZocWait("LOGON:") returns, you can assume that the whole logon-screen has arrived and is ready for input.

See also: ZocWaitIdle, ZocWaitLine, ZocWaitMux, ZocTimeout, ZocReceiveBuf, ZocLastLine, ZocSyncTime, ZocWaitForSeq

ZocWaitForSeq 1|0|"on"|"off"

Normally Wait-commands ignore emulation control sequences in the data stream. If you need to wait for an emulation control, you can use this command to enable their visibility to the Wait commands.

This command also turns on/off filtering of emulation controls for ZocReceiveBuf.

Example:

```
esc= ZocCtrlString("^[")

/* wait for VT220 color reset */
CALL ZocWaitForSeq "On"
Call ZocWait esc||"[0m"
```

See also: ZocWait, ZocWaiMux, ZocReceiveBuf

ZocWaitIdle(<time>)

Wait until there was no data received from the remote host for the given amount of time (in seconds).

If the host keeps sending data, the command will time out after the time set by ZocTimeout (a value of 640 will be returned to indicate the timeout).

Example:

```
CALL ZocTimeout 60
timeout= ZocWaitIdle(2.5)
IF timeout=640 THEN SAY "Host kept sending a steady stream of data for 60
seconds"
ELSE SAY "OK, finally the host did stop the chatter (2.5 seconds of
silence detected)"
```

See also: ZocWait, ZocWaitLine, ZocWaitMux, ZocWaitNumChars, ZocTimeout, ZocSyncTime, ZocReceiveBuf, ZocLastLine

ZocWaitLine()

Wait for the next non empty line of text from the remote host (if you want to wait for the next line, no matter if empty or not, use EXMLPL(ZocWait "^M")).

The received text will then be available using the ZocLastLine function or it can be accessed with some extra coding via ZocReceiveBuf.

If ZocWaitLine times out, it returns a value of 640.

Note: Since REXX is running in its own thread, it is possible that ZocWaitLine will miss lines if new text is received by ZOC while the REXX program is still processing the previously received data. Thus, especially when using ZocWaitLine in a loop, you should just collect the data until it is complete and then process it in a 2nd pass (see the sample below and the description of the ZocSyncTime command).

Example:

```
rc= ZocWaitLine()
IF rc\=640 THEN DO
  reply= ZocLastLine()
  IF reply=="CONNECT" THEN ...
```

Example:

```
/* issue a command that outputs a bunch of lines of data */
CALL ZocSend 'dig emtec.com && echo "<<END>>"^M'

/* first collect all the data and store it in an array */
n= 0
DO FOREVER
    timeout= ZocWaitLine()

    /* exit loop if no more data */
    IF timeout=640 THEN LEAVE

    line= ZocLastLine()

    /* exit loop on a line meeting some condition */
    IF line=="<<END>>" THEN LEAVE

    /* store line in array */
    n= n+1
    data.n= line
  data.0= n
END

/* when done, process each line of collected data in a 2nd pass */
DO i= 1 TO data.0
  line= data.i
```

```
  /* line can now leisurly be processed
     without fear of missing data */
END
```

See also: ZocWait, ZocWaitIdle, ZocWaitMux, ZocWaitNumChars, ZocTimeout, ZocSyncTime

ZocWaitMux(<text0> [, <text1> ...])

Wait for one of multiple texts in the input data stream. The command will return if one of the given texts is found in the incoming data stream or after the default timeout has expired. The return code provides information about which text was found (0, 1, 2, ...) or if the command timed out (return code 640).

Note: The sum of the lengths of all texts must not exceed 4096 characters.

Example:

```
CALL ZocTimeout 45
ret= ZocWaitMux("You have mail", "Main Menu")
SELECT
    WHEN ret=0 THEN CALL handle_maildownload
    WHEN ret=1 THEN LEAVE
    OTHERWISE SIGNAL handle_error
END
```

See also: ZocWait, ZocWaitIdle, ZocWaitLine, ZocWaitNumChars, ZocTimeout, ZocSyncTime

ZocWaitNumChars(<n>)

Waits for a specific number of characters be received. These could be any characters including newline or control characters. To find out which characters were received, set up a receive buffer (see ZocReceiveBuf) before issuing the ZocWaitNumChars command and read its content after the ZocWaitNumChars command returns.

If less than the given number of characters is received within the wait timeout (see ZocTimeout), the function will return the value 640.

Example:

```
CALL ZocTimeout 45
CALL ZocReceiveBuf 100
ret= ZocWaitNumChars(5)
IF ret\=640 THEN DO
   data= ZocReceiveBuf(0)
END
```

See also: ZocReceiveBuf, ZocWait, ZocWaitIdle, ZocWaitLine, ZocTimeout, ZocSyncTime

ZocWindowState(MINIMIZE|MAXIMIZE|RESTORE|ACTIVATE|MOVE:x,y|QUERY)

Set the State of ZOC's main window to the state given or moves the window to the given pixel coordinate (e.g. `MOVE:20,100`).

In used in function call syntax, the command will return new state of the window.

A parameter string of QUERY will just return the current window state as MINIMIZED, MAXIMIZED or RESTORED (please note the extra letter D at the end of those words).

Example:

```
now= ZocWindowState("QUERY")
IF now\="MINIMIZED" THEN DO
  CALL ZocWindowState "MINIMIZE"
END
```

ZocWrite <text>

Write text to screen. This command is similar to REXX's native SAY command, but it does not place the cursor in the next line after printing the text and it understands control codes like `^M` (Enter) or `^[` (ESC).

Example:

```
/* use a VT220 escape sequence to highlight a word */
CALL ZocWrite "Hello ^[[1m World^[[0m"
```

See also: ZocSetScriptOutputDestination

ZocWriteln <text>

Write text to screen and skip to the next line. This command is similar to the REXX SAY command, but it resolves control codes (e.g. `^[` for ESC).

Example:

```
CALL ZocWriteln "Hello ^M^J World"
SAY "Hello"||X2C(0D)||X2C(0A)||"World"
```

See also: ZocSetScriptOutputDestination

Obsolete Functions

ZocAutoConnect
Renamed to ZocConnectHostdirEntry.

ZocAskP
Renamed to ZocAskPassword.

ZocBaud
Replaced by ZocSetDeviceOpts, e.g `CALL ZocSetDeviceOpts "[1]38400-8N1"`.

ZocCaptClr
Replaced by `ZocCommand("CLEARSCROLLBACK")`.

ZocCarrier()
Replaced by ZocGetInfo("ONLINE").

ZocCls)
Renamed to ZocClearScreen.

ZocCursor
Replaced by ZocGetInfo("CURSOR-X") and ZocGetInfo("CURSOR-Y").

ZocDial()
Renamed to ZocConnect.

ZocEndZoc()
Replaced by ZocTerminate.

ZocExec
Renamed to ZocShellExec.

ZocGetFilename(s)
Renamed to ZocAskFilename, ZocAskFilesnames.

ZocGetFolderName
Renamed to ZocAskFolderName.

ZocGetLine
Renamed to ZocWaitLine.

ZocGetPhonebk
Renamed to ZocGetHostEntry.

ZocGetOption
Renamed to ZocGetSessionOption.

ZocGlobal
Renamed to ZocGlobalStore.

ZocLockKeyboard
Replaced by `ZocKeyboard("LOCK")` and `ZocKeyboard("UNLOCK")`.

ZocLoadOpts
Renamed to ZocLoadSessionProfile.

ZocLoadKeyfile
Renamed to ZocLoadKeyboardProfile.

ZocLogging
Renamed to ZocSetLogging.

ZocLogname
Renamed to ZocSetLogfileName.

ZocRestimer
Replaced by `ZocSetTimer("00:00:00")`.

ZocSaveOpts
Renamed to ZocSaveSessionProfile.

ZocScreen
Renamed to ZocGetScreen.

ZocSendBreak
Replaced by `ZocCommand("SENDBREAK")`.

ZocSendKey <number>
Please use ZocGetSessionOption and ZocSend instead.

ZocSetDevParm
Renamed to ZocSetDeviceOpts.

ZocSetDlPath
Please use `CALL ZocSetProgramOption "DownloadPath=<path>"` instead.

ZocSetPhonebk
Renamed to ZocSetHostEntry.

ZocSetEmu
Please use `CALL ZocSetSessionOption "ActiveEmulation=<emu-id>"` or ZocSetEmulation.

ZocSetHost 0|1
Please use `CALL ZocSetSessionOption "Host=yes|no"` instead.

ZocSetOption
Renamed to ZocSetSessionOption.

1.11.11 ZOC-REXX ZocDialog Templates

A template for the ZocDialog command is a text file which lists and defines the elements that appear within dialog window.
The generalized syntax of a dialog template entry is
`<type> <itemid> = "<text>", <val1>, <val2>[, <val3>[, <>val4>]]`
e.g.
`DIALOG IDDLG0 = "Enter Username", 260,260`
The *`type`* of the entry is defined by the first word according to the list below (uppercase required). The *`itemid`* is a user defined string that can be used to retrieve the item's value after the user closes the dialog (e.g. to check the user's input in a text field). The *`text`* depends on the type of item and it is usually either its title or content. Depending on the type of item, *`val1/val2`* are either its x/y location within the window or its size. The values *`val3/val4`* define size (width/height) for those items which have val1/val2 giving a location. Width and height values can sometimes be omitted and to create the item with standard width and height.
The interpreter for these commands is rather basic and unforgiving, so please do not make assumptions about other ways to write the templates beyond the syntax and samples given here and in general do not take syntactical liberties. Strictly one dialog element per line. Lines which start either with `//` or `--` or `#`

are treated as a comment. Empty lines are allowed within the file.

List of possible dialog-template entries:

DIALOG

DIALOG <itemid> = "<title>", <width>, <height>

Defines the window title and size of the dialog.

STATICTEXT

STATICTEXT <itemid> = "<text>", <x-pos>, <y-pos>[, <width>[, <height>]]

Creates a piece of descriptive text (e.g. a label).

EDITTEXT

EDITTEXT <itemid> = "<contents>", <x-pos>, <y-pos>[, <width>[, <height>]]

Creates a field for text entry, preset with the given contents. The itemid can later be used to retrieve the user entry after the dialog was processed.

EDITPASSWORD

Same as EDITTEXT, but with hidden input.

CHECKBOX

CHECKBOX <itemid> = "<title>", <x-pos>, <y-pos>[, <width>[, <height>]]

Creates a checkbox control with the given title in the given location. The itemid can be used to retrieve the state of the checkbox when the dialog was processed. If the text starts with `[*]`, the checkbox will be initially active (checked).

RADIOBUTTON

RADIOBUTTON <itemid> = "<title>", <x-pos>, <y-pos>[, <width>[, <height>]]

Creates a radiobutton control with the given title in the given location. The itemid can be used to retrieve the state of the radiobutton when the dialog was processed. If the text starts with `[*]`, the radiobutton will be initially activated.

GROUPBOX

GROUPBOX <itemid> = "<title>", <x-pos>, <y-pos>, <width>, <height>

Creates a rectangular box within the window to group or separate content. There is a limit of max five groupboxes per dialog.

DROPDOWNLIST

DROPDOWNLIST <itemid> = "<content>", <x-pos>, <y-pos>, <width>, <height>

Creates a multiple-choice field with a list of values where the list of values can be dropped down to select one. The values of the items are provided in the content parameter separated by a vertical bar (e.g. `"Yes|No|Maybe"`). If an item starts with `[*]`, this item will be preselected. The itemid can be used to retrieve name of the selected choice when the dialog was processed. There is a limit of max five dropdownlist items per dialog.

LISTBOX

LISTBOX <itemid> = "<content>", <x-pos>, <y-pos>, <width>, <height>

Creates a list of items where one can be selected. The items are displayed in a vertical list with a possible scrollbar if the listbox is too small to display them all. The values of the items are provided in the content parameter separated by a vertical bar (e.g. `"Yes|No|Maybe"`). If an item starts with `[*]`, this item will be preselected. The itemid can be used to retrieve name of the selected choice when the dialog was processed.

PUSHBUTTON

PUSHBUTTON <itemid> = "<text>", <x-pos>, <y-pos>[, <width>[, <height>]]

Creates a pushbutton, i.e. a button that will close the dialog when pressed. The button which was used to close the dialog, will return a value "1" when queried with the ZocDialog GET command and the itemid.

DEFPUSHBUTTON

Same as PUSHBUTTON but will create a button that is highlighted as the default button (DEFPUSHBUTTON is usually the OK the okay- or finish-button).

Example: test.dlg

```
// Below is the first dialog template within the test.dlg template file
DIALOG MSG1 = "Simple Sample Dialog", 260,140
STATICTEXT ST = "Congratulations!  This is your first dialog.", 12,12, 100,60
DEFPUSHBUTTON P1 = "OK", 120,70

// This is another dialog template within the same template file
DIALOG MAIN = "Sample Dialog", 260,310
STATICTEXT ST1 = "Your name", 12,12, 100
EDITTEXT ED1 = "Harry", 112,10
CHECKBOX CB1 = "I am a programmer", 14,50, 150
GROUPBOX G1 = "Your favorite color?", 12, 90, 200,90
DROPDOWNLIST DD1= "Red|[*]Green|Blue", 20,115, 170,100
RADIOBUTTON RB1 = "VT220", 14,190, 80
RADIOBUTTON RB2 = "[*]Xterm", 100,190, 80
DEFPUSHBUTTON P1 = "OK", 12, 230, 80
PUSHBUTTON P2 = "Cancel", 112, 230, 80
```

Example: Script that uses the above template

```
-- show simple message
CALL ZocDialog "SHOW", "MSG1@test.dlg"

-- show complex dialog and check input
dlgrc= ZocDialog("SHOW", "MAIN@test.dlg")
SAY ">"dlgrc
IF dlgrc=="##OK##" THEN DO
  okpressed= ZocDialog("GET", "P1")
  IF okpressed==1 THEN DO
    SAY "EDITTEXT value: "||ZocDialog("GET", "ED1")
    SAY "CHECKBOX value: "||ZocDialog("GET", "CB1")
    SAY "DROPDOWNLIST value: "||ZocDialog("GET", "DD1")
  END
END
```

1.11.12 Features You May Have Missed

- If you keep the mouse over one of the LEDs in the status bar for some time, a description of the led will appear. This works for the buttons in the toolbar and userbar also.
- Right mouse button on the toolbar opens the toolbar definition dialog.

- ❑ Right mouse button on the userbar opens the userbar definition dialog.
- ❑ Right mouse button on the status bar, opens a popup menu of all options menu choices.
- ❑ Right mouse button on the main window opens a popup menu.
- ❑ With the Alt key pressed you can mark rectangular parts of the screen.
- ❑ Marking one line of text with the mouse while holding the Shift key down, sends that text immediately (set help for options in program settings -> clipboard).
- ❑ Paste-Quoting text with the Ctrl key pressed, reformats the text to fit the screen width.
- ❑ Right mouse button in the host directory opens a popup menu.
- ❑ The quick access option in the host directory's entry edit dialog, puts phone entries into the File menu.
- ❑ The minimize/maximize buttons in the phone dial window switch between a small and large version of that window (also available from host directory dialog options).
- ❑ The snippets window (Options, Window) captures useful stuff (like file names, internet addresses, etc.) for you.
- ❑ The Local Typing field (Alt+C) has a history (cursor up/down).
- ❑ The key map dialog allows different mappings for keys according to the state of the scroll-lock key. This way you could create totally different keyboards and switch between them by use of scroll-lock.

1.11.13 Common Questions (How-To Guide)

Please also check Quick Start Guides for step by step instructions on how to make connections with Telnet, SSH, ISDN etc. and the Trouble Shooting (Problem) Guide.

How do I make connections via serial cable, telnet, SSH, etc?
Please check our Quick Start Guides.

How can I backup or copy the ZOC configuration to a different computer?
To access the folder containing the configuration files, start ZOC, choose File Menu→Show Data Folder.

Then quit ZOC and use the operating system to copy the files from that folder (especially the contents of the Options subfolder) to a thumb drive or to a network computer.

Then start ZOC and choose File Menu→Show Data Folder on the other computer, quit ZOC and then copy the files from the thumb drive to that folder on the new computer.

How can I move the ZOC data folder to a different location?
To access the data folder, choose File Menu→Show Data Folder. Then quit ZOC and move the folder to another location using Windows Explorer or macOS Finder. Next time you start ZOC, the program will ask you to point it to the new location of the data folder.

How do I transfer files to/from a unix host?
You can use Zmodem or SCP. See the help topic at the bottom of the Transfer menu: File Transfer Overview.

How can I change colors and fonts?
These are configured in the session profile (Options→Session Profile→Colors and Options→Session Profile→Layout). Please read also the discussion about Fonts, Window Size, Color.

How do I change the language of the program to English or German?

Add `/LANG:ENGL` or `/LANG:GER` to the commandline (either via the icon or through a commandline.ini file). See ZOC command line.

How can I create an icon for a connection on the desktop?

Please use the Save As function from the File Menu→Quick Connect dialog or edit your host directory entry and use the button in the Shortcuts tab.

How can I change the speed at which pasted text is sent?

This option is in Options→Session Profile→Transfer→Text Sending.

Are there more terminal fonts available?

ZOC searches the system fonts and offers those which have a fixed width. You can add additional fixed fonts to the system (e.g. google for Inconsolata or ProFont) and they will be offered when you restart ZOC.

How do I access a Cisco router which is connected to the serial port?

See quick start for serial connections.

How can I delete stored SSH passwords?

The password store is in Options→Program Settings→Passwords→Stored Passwords.

How can I create a SSH private/pubic keys pair?

There are two options. 1) Make Secure Shell the active connection method (via Options→Session Profile) and choose "Create SSH Key File" from the file menu. Or you can use the commandline ssh-keygen.exe tool, which you will find in the ZOC program folder (the tool is compatible with the OpenSSH ssh-keygen command for which you will find many samples on the internet).

How can I disable the "Are you sure" messages?

These messages are configured under Options→Program Settings→Prompts.

How can I install ZOC on a LAN?

Please check Network and System/Network Administrators

How can I set ZOC as the default telnet app. in Internet Explorer/Firefox/Safari etc.?

Windows: Open the Windows Explorer (not Internet Explorer) and go to View-Menu, Options, File-Types then select URL-Telnet. Then choose Advanced, Open, Edit and set the Application to

```
"C:\Program Files\ZOC9\Zoc.exe" %1
```

macOS: Firefox lets you choose the application to handle SSH: or TELNET: links. Just choose ZOC. To change the default application to handle links in Safari you need a third party application which lets you edit the URL-handlers for macOS. The programs More Internet or RCDefaultApp should both do the trick.

How can I send a Break signal to a Cisco device or router

If you have a telnet or serial connection active, an entry to do this will appear in ZOC's File menu.

How can I set up ZOC for modem tone dialing

Change the modem options and set the dialing command to 'ATDT'. The AT commands are configured in the Serial/Modem options (next to the list where you select the connection type or in the Session Profile).

What is necessary to use the ISDN module with X.25/X.31?

Please go to the connection type options, select ISDN CAPI 2.0 and press the help button. Also, please be aware, that for use with X.31 you probably need to have that feature enabled by your telco.

1.11.14 Common Problems and Questions (Trouble Shooting Guide)

Please also read Quick Start Guides for step by step instructions on how to make connections with Telnet, SSH, ISDN etc. And see the How-To Guide.
If you do not find a solution to a problem below, please visit our web site and ask your question via the support request form.

ZOC tries to access the com port or complains about RTS/CTS when starting

This happens when Serial/Modem is set as your default connection type in the session profile. Please go to Options→Session Profile→Connection Type, then change the connection type to Telnet, SSH or your preferred method of communication and save the session profile.

Telnet connections to IP addresses only start after a delay

ZOC tries to find a matching host name for IP addresses. If you do not have name server entries for your connections, this may take a few seconds. To disable the feature, go to Options→Session Profile→Connection Type→Telnet→Advanced Options and enable No reverse DNS lookup.

The transparency option slows down the computer

In order to make use of the transparency, you need an up-to-date video board. Otherwise the system's processor will have to perform the necessary graphical computations for the transparency effect. If you have an older video board, please disable the option in Options→Session Profile→Window Parts and/or Options→Session Profile→Window.

The selected emulation displays messed up screens

Please make sure that the chosen emulation is suitable for the remote host. Via Telnet and SSH the ZOC emulation is automatically communicated to the remote host, so just try to select a different one. Xterm, VT220 or VT102 will work well on many systems, although some systems require quite specific and sometimes arcane choices.

If you are using another terminal emulator that works well on your system, check its options and see which emulation is used there.

On Linux systems, Yast/MC etc. show strange characters

Please activate the UTF8/Unicode support in the Emulation options for the Linux emulation (try Xterm/Linux/VT220), or choose the UTF8/Unicode character set in Session Profile, Layout.

My emulation basically works, but it shows funny characters instead of line-graphics/boxes

Go to Session Profile, Layout and either try the UTF8/Unicode or the IBM/DOS character set.

Some F-Keys do not work with my remote system

Please make sure that you are using an emulation which supports the full range of F-Keys (e.g. Xterm, VT220, Sun CDE, Ansi SCO or Linux).

Other emulations only support a limited number of f-keys, in that case you will have to map the correct values (if you know what your host expects) for the f-keys yourself in the Options→Keyboard Profiles dialog.

The cursor keys do not work in some VT102/VT220 applications

You should use the gray cursor keys, not the numeric keypad. The keypad is used for other purposes by some remote applications.

Keyboard redefinition does not work

Please note that the keyboard redefinition takes the Num-Lock state into account. Check your Options→Keyboard Profiles and see if your keys are mapped depending on the Num-Lock and Scroll-Lock states.

I cannot connect using Windows-Modems (TAPI)

ZOC does not use Window's Local Settings, which means that you have to type the phone number exactly as you want it to be dialed (including digits to get an outside line or to dial long distance).

ZOC cannot open the COM port while other programs can

Please make sure that the COM port is available by typing `MODE COMx` (where x is the number of the port) at a command prompt.

If this does not result in an error message, make sure that you typed the com port's name correctly in ZOC. Especially do not use space characters between COM and the number. With higher port numbers (above COM10), some systems require the format \\.\COMxx

The error could also appear if the port is currently used by another program.

I keep getting a "No CTS from modem" message

To prevent ZOC and the modem to overrun each others buffers at times, both should be configured to support the so called RTS/CTS handshake. This is especially necessary with transfer speeds of 9600 bps and more.

If you have activated RTS/CTS handshake, ZOC insists on getting the according signal from the modem, because without this signal the communication will be blocked. Hence ZOC issues a warning if the CTS signal from the modem cannot be detected.

If you get this warning, please check that your modem is configured to supply the CTS command (e.g. by using the modem's factory settings), that you are using a 7-wire cable (and adapter) between the modem and the PC and that you are using the correct COM-port in Options→Session Profile→Connection Type and/or in Host Directory→Edit.

If the modem or attached device does not support RTS/CTS, turn off RTS/CTS handshake by clicking the Configure button for Serial/Modem or Serial/Direct.

Zmodem file transfers via telnet fail

Please try different Telnet-options in Options→Session Profile→Connection Type, or Host Directory→Edit→Connection Type→Configure (especially with the CR/NUL options) or run `rz -e` on the remote computer.

Zmodem uploads to the Linux rz command do not start at all or fail

Try to use the -e, -eb or -v option on Unix's side: `rz -e`, `rz -eb` or `rz -v`
If your connection is routed through a Xyplex concentrator, there currently is no known method to make Zmodem uploads at all. You will have to use another protocol -- best choice in this case is probably CKERMIT used from a transfer shell from ZOC (please search the help file for CKERMIT to get details).

Downloads generally work fine, but modem uploads of large files fail

Probably you are using a high-speed modem but have RTS/CTS disabled. For high-speed transmission both, the modem and ZOC, need to be configured to use RTS/CTS handshaking.

After downloads the time setting of files is off by a few hours

Please check the discussion of the TZ environment variable in the starter section of this file or set the 'Ignore time stamp' option in Options→Session Profile→Transfer.

Whenever I leave the program I get a Carrier Detect warning

Maybe your modem is configured to have the CD signal always on. Set it to match the CD state (maybe AT&C1) or set the CD signal to 'invalid' in the serial options.

Sometimes ZOC hangs after starting or when sending to the COM port

Try to disable DSR handshake in the options for Serial/Modem.

My modem works properly with other terminal software but not with ZOC

Try using the same modem initialization as in the other terminal package. Or store that modem configuration into the non-volatile memory of the modem (mostly done with AT&W) and use ATZ as the initialization string for ZOC (AT commands are configured from the Serial/Modem device settings).

ZOC does not show ISDN as a Connection Type under Windows

You need CAPI V2.0 drivers from your board manufacturer, which come with nearly all of the better ISDN cards.

ISDN can be selected, but connections fail

You are probably using a CAPI2032.DLL which does not work with your current drivers, e.g. CAPI2032.DLL from Windows, together with drivers from your board. Also, please set the communication parameters to STANDARD (there is a Standard button in Options→Session Profile→Connection Type→ISDN) and save the options.

Also make sure, that this kind of setting is made in all your ISDN phone book entries. If you still experience problems, please try to reinstall your ISDN drivers and make sure that you are using the latest driver kit from your manufacturer.

No characters appear after making a modem connection

You use a secured transmission and the modem waits to get an OK signal from the computer which does not come. Turn on RTS/CTS in the options for the serial connection.

Sizing the ZOC window only works in large steps

By default ZOC changes the font size when you size the window. Since the font is not available in every necessary size ZOC will use a best-fit method and size the window accordingly. If you do not like this, you can change the sizing behavior in the Options→Session Profile→Window Parts options.

I would like to use more fonts for ZOC

ZOC checks all available fonts in the system to find the ones that could be used for the terminal window. For use in ZOC all characters in the font must have the same width and must support the selected code page.
Under Windows the Courier New, Lucida Console and Terminal fonts meet this requirement. If you do not see the Terminal font in ZOC's font list, please make sure that the 'Show only true type fonts ...' setting in My Computer→Control Panel→Fonts→View→Options→True Type is not selected.

I miss a full screen mode

There is no character based full screen mode available. ZOC is a graphical program and uses the benefits of the graphical user interface (like offering icons). However, you can maximize the ZOC window using the maximize button in the title bar. This way ZOC will cover the whole screen (you should select a large font if you do this).

I disabled the menu, how can I get it back?

You can right-click into the terminal area, this will also bring up the menu. If you mapped another function on right-click, you will need to manually edit the session profile (e.g. My Documents → ZOC9 Files → Options → Standard.zoc) using a text editor.

Scrolling is slow

Scrolling in an window is generally slower than in full screen mode. There is a scroll speed option in Options→Session Profile→Terminal.

Is there a host mode?

Yes. It is implemented as a REXX program. To start it select Script, Start Script and select MINIHOST.ZRX. The program will prompt you for two passwords (for guest and supervisor mode) and will then wait for incoming calls.

What does ZOC mean?

The Hitchhikers Guide to the Galaxy might describe ZOC as followed:
zoc (v), to `zoc` means communicating with other people through the means of extremely sophisticated hard- and software (sophisticated at least from the viewpoint of the ape-descended inhabitants of an extremely unimportant blue planet in an even lesser important part of the galaxy who think that personal computers are a pretty neat pieces of hardware).