
1.1 ZOC Überblick

Copyright (C) 1993 - 2025

Was ist ZOC Terminal

ZOC ist eine professionelle Terminalemulation mit der Sie zeichenorientierte Gegenstellen via SSH, Telnet, Modem, ISDN erreichen können. ZOC wird verwendet um auf Unix/Linux oder IBM3270 Servern zu arbeiten, oder mit anderen Rechnern Daten auszutauschen.

Neben weit verbreiteten Terminalemulationen wie Linux, VT102, VT220 und verschiedenen Ansi Varianten werden auch Emulationen wie IBM3270 oder Sun CDE unterstützt sowie Dateiübertragung per X-, Y- Zmodem sowie SCP und Kermit.

ZOC ist hochgradig konfigurierbar und bietet auch Features wie Aufbau mehrerer Verbindungen in einem Fenster (Tabs), eine mächtige Skriptsprache, automatische Aktionen bei bestimmten empfangenen oder eingegebenen Texten, Verbindungsprotokolle zur Fehlersuche und vieles mehr.

Einstieg nach Einsatzgebieten

- ☐ [Secure Shell \(SSH\) Verbindungen](#)
- ☐ [Telnet Verbindungen](#)
- ☐ [Direkte serielle Verbindungen](#)
- ☐ [Modem Verbindungen](#)
- ☐ [Skripts/Programmierung/REXX](#)
- ☐ [System/Netzwerk Administratoren \(Installation und Verteilung\)](#)

Weitere Überblicksthemen

- ☐ [ZOC Kommandozeile](#)
- ☐ [Bildschirmelemente](#)
- ☐ [Das ZOC Menü \(Funktionen\)](#)
- ☐ [Bestellungen/Updates/Kontakt](#)
- ☐ [Häufige Fragen \(Wie geht das?\)](#)
- ☐ [Häufige Probleme und Fragen](#)
- ☐ [Leicht zu übersehende Funktionen](#)
- ☐ [ZOC Deinstallation](#)

1.1.1 Einstieg nach Einsatzgebieten

ZOC ist eine professionelle Terminalemulation mit der Sie zeichenorientierte Gegenstellen via SSH, Telnet, Modem, usw. erreichen können.

Unten finden Sie eine Liste von Beschreibungen zum schnellen Einstieg in ZOC, gegliedert nach Einsatzgebiet.

- ☐ [Secure Shell \(SSH\)](#)
- ☐ [Telnet](#)
- ☐ [Direkte serielle Verbindungen](#)
- ☐ [Modem](#)
- ☐ [Skripts/Programmierung](#)
- ☐ [System/Netzwerk Administratoren \(Installation und Verteilung\)](#)

1.1.1.1 *Secure Shell (SSH) Einstieg*

Verbindungsaufbau

Um eine Verbindung über [Secure Shell \(SSH\)](#) herzustellen, wählen Sie im Menü [Dateimenü→Neue Verbindung](#).

Im daraufhin angezeigten Dialog kann die Gegenstelle, die gewünschte Emulation ([VT220](#) oder [Xterm](#) funktionieren in den meisten Fällen) sowie [Secure Shell](#) als Verbindungstyp ausgewählt werden. Geben Sie unter [Verbinden mit](#) den Namen der Gegenstelle bzw. die IP-Adresse an, also z.B.

[ssh.hogwarts.edu](#) oder [192.168.1.1](#)

Wenn der Server, zu dem Sie sich verbinden möchten, einen anderen Port als 22 (dem üblichen SSH-Port) verwendet, können Sie eine alternative Portnummer im dafür vorgesehenen Port-Eingabefeld angeben. Wird das Feld für den Port leer gelassen, wird der Standardport verwendet.

Wenn Sie sonstige Terminal- und Fenstereinstellungen ändern möchten, klicken Sie den Knopf [Bearbeiten](#) für das [Verbindungsprofil](#). (Verbindungsprofile sind vordefinierte Sammlungen von Einstellungen die den einzelnen Verbindungen zugrunde liegen.)

Direktwahl/Icons

Beachten Sie auch [Speichern unter](#) Knopf. Hiermit können Sie z.B. ein Icon für eine Verbindung auf der Arbeitsoberfläche erstellen oder die Verbindung ins Adressbuch eintragen.

Adressbuch

Um eine Secure Shell Gegenstelle direkt im Adressbuch anzulegen, öffnen Sie [Dateimenü→Adressbuch](#), klicken Sie auf [Neu](#).

Dann geben Sie den Namen des Eintrags und den Hostnamen oder die Adresse der Gegenstelle an und wählen Sie als Verbindungstyp [Secure Shell \(SSH\)](#). Im Adressbucheintrages kann außerdem eine Emulation und weitere Optionen für den Eintrag eingestellt werden.

Wenn Sie möchten, dass sich ZOC am Host automatisch einloggt, können Sie den Benutzernamen und das Passwort im Abschnitt [Login](#) hinterlegen.

Sicherheitswarnung

Das Speichern des Passwortes gibt jedem Benutzer, der Zugriff auf ZOC hat, die Möglichkeit, sich mit dieser Gegenstelle zu verbinden, ohne das Passwort eingeben zu müssen. Bitte verwenden Sie diese Option mit Vorsicht.

Dateitransfer

Um Dateien per Secure Shell (SSH) zu übertragen, lesen Sie bitte die Beschreibung zum [Secure Shell File-Transfer](#).

Verwandte Themen

Mehr über [Secure Shell \(SSH\)](#) (z.B. Optionen, Hostnamen, Ports, etc.). Außerdem können Sie Secure Shell (SSH) Verbindungen mit dem /SSH Parameter über die [Kommandozeile](#) aufbauen. Darüber hinaus lesen Sie bitte die Hilfe im Dialog beim Anlegen eines Telefonbucheintrags.

1.1.1.2 Telnet Einstieg

Verbindungsaufbau

Um eine Verbindung über [Telnet](#) herzustellen, wählen Sie im Menü [Dateimenü→Neue Verbindung](#).

Im daraufhin angezeigten Dialog kann die Gegenstelle, die gewünschte Emulation ([VT220](#) oder [Xterm](#) funktionieren in den meisten Fällen) sowie Telnet als Verbindungstyp ausgewählt werden. Geben Sie unter [Verbinden mit](#) den Namen der Gegenstelle bzw. die IP-Adresse an, also z.B. [bbs.fonix.org](#) oder [192.168.1.1](#)

Wenn der Server, zu dem Sie sich verbinden möchten, einen anderen Port als 23 (dem üblichen Telnet-Port) verwendet, können Sie eine alternative Portnummer im dafür vorgesehenen Port-Eingabefeld angeben. Statt der Portnummer können Sie auch einen Servicenamen verwenden, z.B. [smtp](#). Wird das Feld für den Port leer gelassen, wird der Standardport verwendet.

Wenn Sie sonstige Terminal- und Fenstereinstellungen ändern möchten, klicken Sie den Knopf [Bearbeiten](#) für das [Verbindungsprofil](#). (Verbindungsprofile sind vordefinierte Sammlungen von Einstellungen die den einzelnen Verbindungen zugrunde liegen.)

Direktwahl/Icons

Beachten Sie auch [Speichern unter](#) Knopf. Hiermit können Sie z.B. ein Icon für eine Verbindung auf der Arbeitsoberfläche erstellen oder die Verbindung ins Adressbuch eintragen.

Adressbuch

Um eine Telnet-Gegenstelle direkt im Adressbuch anzulegen, öffnen Sie [Dateimenü→Adressbuch](#), klicken Sie auf [Neu](#).

Dann geben Sie den Namen des Eintrags und die IP-Adresse der Gegenstelle an und wählen Sie als Verbindungstyp [Telnet](#). Im Adressbucheintrages kann außerdem eine Emulation und weitere Optionen für den Eintrag eingestellt werden.

Mit der Funktion [AutoLogin bei der nächsten Anmeldung aufzeichnen](#) kann ZOC den Anmeldevorgang aufzeichnen und später automatisiert ausführen.

Verwandte Themen

Mehr über [Telnet](#) (z.B. Optionen, Hostnamen, Ports, etc.). Außerdem können Sie Telnet Verbindungen mit dem /TELNET Parameter über die [Kommandozeile](#) aufbauen. Unter [Häufige Fragen](#) finden Sie eine Anleitung um ZOC als Standard Telnet-Client in Ihren Webbrowser einzutragen. Darüber hinaus lesen Sie bitte die Hilfe im Dialog beim Anlegen eines Telefonbucheintrags.

1.1.1.3 Direkte Serielle Verbindungen Einstieg

Verbindungsaufbau über die Funktion 'Neue Verbindung'

Um Verbindungen mit Geräten (z.B. Router oder Microcontroller) aufzubauen, die direkt an einem seriellen Port angeschlossen sind, wählen Sie im Menü [Dateimenü→Neue Verbindung](#).

In diesem Fenster wählen Sie [Serial/Direct](#) als Verbindungstyp und klicken den [Konfigurieren](#)-Knopf. Stellen Sie dort den COM Port und die seriellen Parameter ein.

Im Feld [Verbinden mit](#) können Sie dann entweder den Text `localhost` oder `serial` angeben. Damit wird der COM Port angesprochen, der mit Hilfe des [Konfigurieren](#)-Knopfes eingestellt wurde.

Alternativ können Sie im Feld [Verbinden mit](#) den Namen eines seriellen Ports angeben. Dieser Port wird dann statt des Ports verwendet, der mit dem [Konfigurieren](#)-Knopf eingestellt wurde.

Um gültige Bezeichnungen für die Ports zu finden, klicken Sie auf [Scan](#) im [Konfigurieren](#) Dialog.

Verbindungsaufbau per Adressbuch

Darüber hinaus können Sie auch das Adressbuch (Dateimenü) für solche Verbindungen nutzen. Wählen Sie für einen Eintrag dort auch [Serial/Direct](#) als Verbindungstyp. Wie oben bei 'Neue Verbindung' beschrieben, kann auch hier das Feld [Verbinden mit](#) entweder `localhost` oder `serial` oder den Namen eines seriellen Ports enthalten.

Unterschied zwischen Serial/Direct und Serial/Modem

Auf den ersten Blick erscheinen die Verbindungstypen [Serial/Direct](#) und [Serial/Modem](#) sehr ähnlich. Der entscheidende Unterschied ist wie folgt:

[Serial/Direct](#) dient dazu, mit einem *lokalen Gerät* (Router, Microcontroller, etc.) zu kommunizieren, das direkt an einen seriellen Port des Rechners angeschlossen ist.

[Serial/Modem](#) hingegen dient dazu, mit einem *entfernten Rechner* zu kommunizieren. Das am seriellen Port angeschlossene Gerät (Modem) dient hier lediglich dem Verbindungsaufbau und ist nicht das primäre Ziel der Kommunikation.

Verwandte Themen

Mehr über die Verbindungsmethoden [Serial/Direct](#) und [Serial/Modem](#) finden Sie unter [Serial/Direct](#).

1.1.1.4 Modem Einstieg

Verbindungsaufbau

Um eine Verbindung über [Modem oder ISDN-Modem](#) herzustellen, wählen Sie im Menü [Dateimenü→Neue Verbindung](#).

Im daraufhin angezeigten Dialog wählen Sie bitte entweder [Serial/Modem](#) (wenn Sie auf das Modem direkt über den COM-Port zugreifen möchten) oder [Windows Modems](#).

Ändern Sie Optionen wie z.B. den COM-Port durch Drücken der [Konfigurieren](#)-Schaltfläche neben der Auswahl des Verbindungstyps.

Geben Sie unter [Verbinden mit](#) die Telefonnummer der Gegenstelle an und wählen Sie auf Wunsch eine Bildschirmemulation aus (`vt220` oder `ANSI` funktioniert in den meisten Fällen).

Wenn Sie sonstige Terminal- und Fenstereinstellungen ändern möchten, klicken Sie den Knopf [Bearbeiten](#) für das [Verbindungsprofil](#). (Verbindungsprofile sind vordefinierte Sammlungen von Einstellungen die den einzelnen Verbindungen zugrunde liegen.)

Direktwahl/Icons

Beachten Sie auch [Speichern unter](#) Knopf. Hiermit können Sie z.B. ein Icon für eine Verbindung auf der

Arbeitsoberfläche erstellen oder die Verbindung ins Adressbuch eintragen.

Adressbuch

Um eine Modem-Gegenstelle direkt im Adressbuch anzulegen, öffnen Sie [Dateimenü→Adressbuch](#), klicken Sie auf [Neu](#). Dann geben Sie den Namen des Eintrags und die Telefonnummer der Gegenstelle an und wählen Sie als Verbindungstyp [Serial/Modem](#) oder [Windows-Modem](#).

Ändern Sie auch die Einstellungen des Verbindungstyps (wie z.B. den COM-Port des Modems) durch Klick auf die [Konfigurieren](#)-Schaltfläche neben dem Verbindungstyp.

Im Adressbucheintrages kann außerdem eine Emulation und weitere Optionen für den Eintrag eingestellt werden.

Mit der Funktion [AutoLogin bei der nächsten Anmeldung aufzeichnen](#) kann ZOC den Anmeldevorgang bei der Gegenstelle aufzeichnen und später automatisiert ausführen.

Unterschied zwischen Serial/Direct und Serial/Modem

Auf den ersten Blick erscheinen die Verbindungstypen [Serial/Direct](#) und [Serial/Modem](#) sehr ähnlich. Der entscheidende Unterschied ist wie folgt:

[Serial/Direct](#) dient dazu, mit einem *lokalen Gerät* (Router, Microcontroller, etc.) zu kommunizieren, das direkt an einen seriellen Port des Rechners angeschlossen ist.

[Serial/Modem](#) hingegen dient dazu, mit einem *entfernten Rechner* zu kommunizieren. Das am seriellen Port angeschlossene Gerät (Modem) dient hier lediglich dem Verbindungsaufbau und ist nicht das primäre Ziel der Kommunikation.

Verwandte Themen

Details über den Modem-Zugriff finden Sie in [Serial/Modem](#) und [Windows Modems](#). Bei Verwendung von [Serial/Modem](#) als Verbindungstyp finden Sie weitere Informationen über die möglichen Einstellungen unter [Modem-Einstellungen](#).

1.1.1.5 Einstieg in Skripts/Programmierung

Falls Sie ZOC mit Hilfe der Skriptsprache REXX automatisieren wollen, finden Sie mehr hierüber in

- ☐ [Einführung in die ZOC REXX Programmierung](#) (empfohlen!)
- ☐ [REXX Sprachelemente](#)
- ☐ [ZOC-REXX Befehle](#)

Diese Themen sind auch direkt über das Hilfemenü im Hauptfenster erreichbar.

Sie sollten auch die Dokumentation und die Beispiele (vor allem das Tutorial) im ZOC Datenordner ([Dateimenü→Datenordner anzeigen](#)) und die REXX Ressourcen auf <https://www.emtec.com/zoc/documents.htm#rexxfiles> nicht übersehen.

Falls Sie Skripts von der Kommandozeile aus aufrufen wollen, lesen Sie bitte den Abschnitt über [Kommandozeilenparameter](#) und dort speziell die Teile `/RUN:` und `/U`.

Falls Sie ZOC zur Realisierung von komplexen Problemen verwenden, können Sie auch gern mit uns [Kontakt aufnehmen](#).

Als Alternative zur REXX könnte [DDE \(Dynamic Data Exchange, Windows\)](#) oder [AppleScript \(macOS\)](#) für Sie interessant sein.

1.1.1.6 **Installation und Verteilung für System- und Netzwerkadministratoren**

Wenn Sie System- oder Netzwerkadministrator sind und ZOC für die Benutzer in Ihrer Firma installieren und vorkonfigurieren, sind die nachfolgenden Punkte sicher hilfreich für Sie.

Falls Sie darüber hinaus Informationen benötigen und spezielle Anforderungen haben, kontaktieren Sie bitte unseren Support.

Kommandozeilenparameter

ZOC bietet eine Vielzahl von Parametern, um in speziellen Modi zu starten oder beim Starten automatisch bestimmte Funktionen auszuführen. Beim Lesen der [Kommandozeilenoptionen](#) achten Sie bitte besonders auf die Optionen `/CALL`, `/RUN`, `/RESTRICT`, `/U` und `/WD`. Diese Parameter können direkt in die Startverknüpfung von ZOC eingetragen oder in einer Datei namens [commandline.ini](#) hinterlegt werden (letztere kann auch ein Teil einer vorkonfigurierten Installation sein, siehe unten).

Konfigurationsdateien

Wenn Sie eine ZOC-Konfiguration in der gesamten Firma replizieren wollen, werden hierzu die Dateien [HostDirectory.zhd](#) (Adressbuch), [Standard.zfg](#) (Programmooptionen), [*.ZOC](#) (Verbindungsprofile), [*.ZKY](#) (Tastaturbelegungen) und eventuell die [*.ZRX](#) (REXX Skript) Dateien benötigt. Diese liegen im Datenordner von ZOC (siehe [Datei-Menü→Datenordner anzeigen](#)).

Außerdem dürfte die Datei [Admin.ini](#) und der Ordner [newuserprofile_german](#) (im Programmordner) von besonderem Interesse sein.

Lizenzierung

Normalerweise werden Lizenzcodes in der Windows-Registry unter HKLM/HKCU+Software+EmTec+ZOC9+Registration hinterlegt. Alternativ bietet ZOC aber auch eine dateibasierte Lizenzierung sowohl eine netzwerkbasierte Lizenzüberprüfung für Einzel- und Netzwerk-Installationen (siehe unten) an. Bitte wenden Sie sich an den [EmTec Support](#) für weitere Details).

Netzwerkinstallation/Multi-User Installation

ZOC kann auf einem Netzlaufwerk installiert werden und kann mit gemeinsamen oder individuellen Konfigurationen sowohl mit als auch ohne Schreibberechtigung eingesetzt werden. Weiterhin kann ZOC anhand der [Admin.ini](#) Datei (Beispiel dafür im ZOC-Programmverzeichnis) und anhand der Einstellungen in den Programmooptionen, Adressbuchdateien und Optionsdateien aus beliebigen Verzeichnissen verwenden (siehe auch [Ordner für Benutzerdaten](#)). Zusätzlich können [Kommandozeilenoptionen](#) evtl. auch manche Kommandozeilenparameter hierfür nützlich sein, wobei wir die Nutzung der Admin.ini Datei empfehlen.

Auf Wunsch berät Sie der [EmTec Support](#) gerne bezüglich der passenden Netzwerk-Installation und dem entsprechenden Lizenzcode-Handling für Ihre Firma.

Erstellen eines benutzerdefinierten Setups

Die ZOC-Installation kann gemäß Ihren Anforderungen angepasst werden. Die Installationsdatei (zoc_nnn.exe) ist ein selbstentpackendes ZIP-Archiv. Dieses Archiv kann mit jedem UNZIP-Programm in ein temporäres Verzeichnis entpackt werden (z.B. Z:\CDINST).

Dieses Verzeichnis enthält unter anderem die beiden Dateien [SETUP.CFG](#) und [SETUP.FIL](#). Die Datei [SETUP.CFG](#) enthält das Setup-Skripts und steuert den Standard-Verzeichnisnamen sowie die Icons im Windows-Startmenü. [SETUP.FIL](#) ist wiederum ein ZIP-Archiv, welches das gesamte ZOC-Verzeichnis und alle Standard-Konfigurationsdateien enthält. Diese Standard-Optionsdateien (im Unterordner [newuserprofile_german](#)) können auf Wunsch ersetzt sowie weitere Konfigurationsdateien hinzugefügt werden.

Alternativ kann ein zusätzliches ZIP-Archiv mit dem Namen [CUSTOM.FIL](#) erzeugt werden, dass während

der Installation direkt nach dem Entpacken der Dateien aus dem SETUP.FIL Archiv nachverarbeitet wird. Das derart veränderte Verzeichnis Z:\CDINST kann dann auf eine CD gebrannt oder als LAN-Ressource den lizenzierten Benutzern im Unternehmen zur Verfügung gestellt werden. In beiden Fällen startet die modifizierte Installation mit dem Ausführen von SETUP.EXE.

Unbeaufsichtigte Installation

Das ZOC-Setup bietet keine Installation ohne Benutzereingriff an. Allerdings kann das Verzeichnis eines bereits installierten ZOC ohne Probleme auf andere Rechner repliziert werden. Die Icon-Erstellung und Lizenzierung müsste dann allerdings mit den externen Setup-Tools realisiert werden.

Fragen?

Wenn Sie weiterführende Fragen zur Installation oder zur Lizenzierung von ZOC haben, werden Sie sich bitte an Markus Schmidt von [EmTec](#).

1.1.1.7 ZOC Deinstallation

Windows

Die Windows-Version von ZOC enthält ein Programm zur Deinstallation. Die Deinstallation kann entweder über den entsprechenden Eintrag in der ZOC Programmgruppe im Startmenü, oder aber über den Eintrag unter Software in der Systemsteuerung von Windows gestartet werden.

Wenn Sie die ZOC-Dateien manuell entfernen möchten, löschen Sie bitte den ZOC-Programmordner [C:→Programme→ZOC9](#), das ZOC-Datenverzeichnis [Meine Dokumente→ZOC9 Dateien](#) und die Registry-Einträge [Software→EmTec, ZOC9](#) in HKEY_CURRENT_USER und HKEY_LOCAL_MACHINE.

macOS

Unter macOS werden das Programm und ZOC-Konfigurationsdateien in folgenden Verzeichnissen abgespeichert:

[/Applications/zoc9.app](#)

[/Library/Preferences/com.emtec.*](#)

[~/Library/Preferences/com.emtec.*](#)

[~/Library/Application Support/ZOC*Files/](#)

[~/Library/Logs/EmTec*](#)

Durch eine Suche nach [ZOC](#) im Finder oder Spotlight werden diese Dateien ebenfalls angezeigt.

1.2 ZOC starten (Kommandozeilenparameter und SET-Variablen)

Themen in diesem Abschnitt:

- ☐ [Kommandozeilenparameter](#)
- ☐ [Kommandozeilendateien \(commandline.ini, *.zsh/.zln\)](#)
- ☐ [ADMIN.INI, Arbeitsverzeichnis, Netzwerk](#)
- ☐ [SET-Variablen](#)
- ☐ [Autostart REXX-Programme](#)

1.2.1 Kommandozeilen-Parameter

ZOC kennt eine Vielzahl von Parametern, die beim Starten entweder von der Kommandozeile, vom Programmsymbol (Windows Icon) oder durch [Kommandozeilendateien \(Windows, macOS\)](#) übergeben werden können.

Hinweis: Zusätzlich zum Aufruf von ZOC mit den unten aufgelisteten Parametern können Sie die Kommandozeilentools `ssh.exe` und `telnet.exe` verwenden, die im ZOC Programmordner gespeichert sind. Diese Programme akzeptieren Parameter, die kompatibel mit OpenSSH und Windows Telnet sind, und wandeln diese dann in einen Aufruf mit den ZOC Parametern um. Für weitere Informationen starten Sie diese Programme bitte in einem Eingabefenster mit `-?`

```
ZOC
[/CALL:<hostname>] [/CONNECT:<verbindungstyp>!<host>]
[/DEV:<verbindungstyp>] [/EMU:<emulation>]
[/FI:<zeichensatz>] [/FX:<zeichensatz>]
[/FORCEASIANFONTS] [/FORCENOASIANFONTS]
[/KEY:<datei>]
[/LANG:<sprache>]
[/MIN] [/MAX] [/MOUSEPOS:<x>,<y>] [/NUMLOCK:<n>]
[/O:<verbindungsprofil>]
[/HOSTDIR:<adressbuchdatei>]
[/PROGCFG:<programmconfig>]
[/POS:<x>,<y>,<breite>,<höhe>]
[/RESTRICT:n]
[/RUN:<scriptdatei>] [/RUNARG:<scriptargument>]
[/RLOGIN:<host>[:<port>]]
[/SSH:<login>@<host>[:<port>]] [/SSHKEY:<private keyfile>]
[/STANDALONE]
[/TABBED]
[/TELNET:<host>[:<port>]]
[/TT:<datei>] [/U] [/WD:<arbeitsverz.>]
```

Parameter können beim Starten von ZOC aus einem Windows Befehlseingabefenster oder in einem Verknüpfungs-Icon angegeben werden. Um z.B. ZOC mit einer Telnet-Verbindung zu starten, klicken Sie die rechte Maustaste auf der Arbeitsoberfläche, wählen Neu, Verknüpfung und geben `"C:\Programme\zoc9\zoc.exe" /TELNET:mailserver.de` an.

Bitte beachten Sie, dass Parameter, die ein Leerzeichen enthalten, in Hochkommata eingeschlossen werden müssen, z.B.

```
"C:\Programme\zoc9\zoc.exe" "/CALL:Neuer Mail Server" oder
zoc "/EMU:Ansi BBS"
```

Normalerweise wird das aktuelle Arbeitsverzeichnis beim Start von ZOC ignoriert, (siehe [Datenverzeichnis](#)) aber Sie können in Kommandozeilenparametern den Platzhalter `%CURDIR%` verwenden, um sich auf das Startverzeichnis zu beziehen, z.B. `/O:%CURDIR%\test.zoc`

Alternativ zu dem oben gezeigten Windows typischen Format der Kommandozeilenparameter ist es möglich, die Parameter im Unix-Stil anzugeben und statt des Schrägstrichs ein Minus zu verwenden. Außerdem kann in Unix-Schreibweise der Doppelpunkt für Wertangaben gegen ein Leerzeichen ersetzt werden. Demgemäß führen die folgenden Aufrufe zum gleichen Ergebnis:

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

Beschreibung der einzelnen Befehlszeilenparameter

/CALL:hostentry

Hier kann der Name eines Adressbucheintrages angegeben werden, der sofort nach Programmstart automatisch ausgeführt wird, z.B. `ZOC /CALL:CoolServer`

/CONNECT=<verbindungstyp>![<host>

Diese Option erlaubt es, von der Befehlszeile aus eine beliebige Gegenstelle mit einer beliebigen Verbindungsmethode anzuwählen.

Als Verbindungstyp-Parameter muss einer der Namen der ZOC-Verbindungstyp-Optionen angegeben werden, für Host kann ein Hostname, eine Adresse oder eine Telefonnummer (abhängig vom verwendeten Verbindungstyp) stehen (bei Hostnamen kann zusätzlich mit Doppelpunkt noch ein Port angehängt werden).

Bei Verbindungen per SSH (Secure Shell) kann ein Benutzername und Passwort in der Form `<benutzername>:<passwort>@<host>` übergeben werden, also z.B. `ZOC /CONNECT=SSH!harry:alohomora@secure.com`. (Zur Anmeldung mit Public/Private-Key Dateien lesen Sie bitte den Text zum `SSH` und `SSHKEY` Parameter.)

Für `RLOGIN` kann `<benutzername>@<host>` verwendet werden.

/DEV:verbindungstyp

Mit `/DEV` kann angegeben werden, welchen Verbindungstyp ZOC beim Starten aktivieren soll. Nach dem Doppelpunkt muss der Name (wie in [Verbindungsprofil](#)→[Verbindungstyp](#) angezeigt) angegeben werden, z.B. `ZOC /DEV:TELNET`

Falls nötig, können Verbindungstyp-Parameter (wie im REXX Kommando `ZocSetDevParm` beschrieben) mit `@` an den Namen des Verbindungstyps angehängt werden, z.B. `ZOC "/DEV:SERIAL/MODEM@COM3:57600-8N1|9|350"` (Parameter, die das `|` Zeichen enthalten, müssen in Hochkommata eingeschlossen werden).

/EMU:emulation

Mit diesem Parameter geben Sie an, welche Terminalemulation nach dem Start aktiviert werden soll, z.B. `ZOC /EMU:VT220`

/FI

Normalerweise überprüft ZOC alle im System vorhandenen Zeichensätze, um zu überprüfen, welche

für ZOC geeignet sind. Diese Liste kann deshalb Zeichensätze enthalten, die unerwünscht sind. Mit der Option /FI verwendet ZOC nur Zeichensätze, die explizit angegeben werden, der Aufruf `ZOC "/FI:Courier New"` hat zur Folge, dass ZOC in der Liste der Zeichensätze nur noch Courier New anbietet. Diese Option kann mehrfach verwendet werden, um mehrere Zeichensätze einzubinden. Es ist wichtig, dass zumindest einer der angegebenen Zeichensätze wirklich existiert und für ZOC nutzbar ist, ansonsten bricht ZOC beim Starten mit einer Panic-Fehlermeldung ab.

/FX

Die /FX Option funktioniert ähnlich der Option /FI, allerdings verwendet ZOC alle normalerweise verwendeten Zeichensätze mit Ausnahme der hier angegebenen. Um also alle Terminal-Zeichensätze auszuschließen, lautet der Aufruf `ZOC /FX:Terminal`.

/FORCE[NO]ASIANFONTS

Unter Windows ignoriert normalerweise ZOC spezifische asiatische Schriftarten wie SimSun oder BatangChe wenn die Systemsprache English oder Deutsch ist. Mit dieser Option, werden die asiatischen Schriftarten immer bzw. nie angeboten.

/KEY:datei

Diese Option bewirkt, dass ZOC beim Laden eine spezielle Tastaturdefinitionsdatei lädt, z.B. `ZOC /KEY:Alternate.zky`

/LANG:GER|ENGL

Mit dieser Option lässt sich die Sprache des Programms zwischen deutsch (`/LANG:GER`) und englisch (`/LANG:ENGL`) wechseln.

/MIN

Diese Option bewirkt, dass ZOC beim Starten mit minimiertem Fenster geöffnet wird.

/MAX

Diese Option bewirkt, dass ZOC beim Starten mit maximiertem Fenster geöffnet wird.

/MOUSEPOS

Bewegt die Maus nach dem Start an die angegebene Pixelposition z.B. `/MOUSEPOS:1024,768` (nur Windows)

/NUMLOCK:n

Mit dieser Option kann eingestellt werden, ob ZOC mit eingeschalteter (n=1) oder ausgeschalteter (n=0) Num-Taste gestartet wird. (nur Windows)

/O:datei

Mit dieser Option lädt ZOC beim Starten das angegebene Verbindungsprofil (statt `OPTIONS\STANDARD.ZOC`), z.B. `ZOC /O:BlackGlass.ZOC`

/PROGCFG:datei

Mit dieser Option kann der Name einer Datei mit Programmeinstellungen angegeben werden, die statt der `Standard.zfg` beim Programmstart geladen wird.

/HOSTDIR:datei

Hier kann der Name einer Adressbuchdatei angegeben werden, die statt des Standard-Adressbuchs `HostDirectory.zhd` beim Programmstart geladen wird.

/RESTRICT:n

Mit dieser Option kann der Zugriff auf verschiedene Teile von ZOC eingeschränkt werden. Diese Option steht jedoch nur noch aus Kompatibilitätsgründen zur Verfügung, bitte nutzen Sie stattdessen die `ADMIN.INI` Datei (Beispieldatei im ZOC-Verzeichnis).

/RUN:scriptname

Mit dieser Option kann ZOC ein REXX-Programm übergeben werden, das nach dem Starten des Programms automatisch ausgeführt wird, z.B. `ZOC /RUN:PROCESS.ZRX`.

Das Programm wird im eingestellten REXX-Ordner ([Optionen→Programmeinstellungen→Ordner](#)), im ZOC Programm-Ordner oder im Benutzerdatenverzeichnis (ZOC9 Dateien) gesucht. Alternativ kann ein Pfad angegeben werden, z.B. `ZOC "/RUN:C:\MY SCRIPTS\NIGHTCALL.ZRX"`

Sollte das REXX-Programm eine Parameterübergabe erwarten (diese wird im Skript als `parm= ARG(1)` abgerufen), kann dieser mit Hilfe von `/RUNARG:` übergeben werden: `ZOC /RUN:DOIT.ZRX
"/RUNARG:HELLO WORLD"`.

Hinweis: Der `/RUN` Parameter kann nicht zusammen mit Parametern verwendet werden, die eine Verbindung aufbauen (z.B. `/CONNECT`, `/TELNET` etc.). In solchen Fällen verwenden Sie bitte nur den `/RUN` Parameter und bauen die Verbindung vom Skript aus mit `Call ZocConnect` etc. auf.
Hinweis: Bitte auch die Option `/U` beachten.

`/RLOGIN:`

Entspricht `/CONNECT=RLOGIN!<benutzer>@<host>:<port>` (siehe oben), z.B. `ZOC /RLOGIN:harry@secure.hogwarts.edu`

`/STANDALONE`

Unter Windows bewirkt dieser Parameter, dass ein ZOC-Fenster getrennt von anderen ZOC-Fenstern geöffnet wird. Technisch gesehen bedeutet dies, dass das Fenster und dessen Sitzungen in einem eigenen Prozessraum laufen (d.h. der Task-Manager wird eine `zoc.exe` je Fenster anzeigen). In der Praxis ergibt sich der Vorteil, dass, sollte ein ZOC-Fenster abstürzen, dies keine Auswirkung auf weitere ZOC-Fenster hat. Gleichzeitig werden aber einige Funktionalitäten eingeschränkt. So ist es z.B. nicht mehr möglich, Registerblätter zwischen ZOC-Fenstern hin- und herzuziehen.

Hinweis: Um alle ZOC-Instanzen in eigenen Prozessen zu starten, bietet es sich an, diesen Parameter über die Datei [commandline.ini](#) zu setzen.

`/TABBED`

Dieser Parameter funktioniert in Verbindung mit anderen Parametern wie `/CONNECT`, `/TELNET`, `/SSH`, oder `/RUN` und erlaubt es, eine Verbindung oder ein Skript als Registerblatt in einem bereits geöffneten ZOC Fenster zu starten, statt für jeden Aufruf ein eigenes Fenster zu öffnen. In Verbindung mit diesem Parameter können Sie noch `/TITLE:<name>` angeben um einen Titel für das Registerblatt festzulegen.

`/SSH:ziel`

Entspricht `/CONNECT=SSH![<benutzer>[:<password>]]@<host>:<port>` (siehe oben), z.B.
`ZOC /SSH:harry@secure.hogwarts.edu`
`ZOC /SSH:harry:alohomora@secure.hogwarts.edu:10022`
`zoc -ssh harry@secure.hogwarts.edu`
`zoc -ssh harry@secure.hogwarts.edu 10022.`

Wenn der Benutzername oder das Passwort Sonderzeichen enthält, die mit dieser Syntax in Konflikt stehen, verwenden Sie die Parameter `/SSHUSER`, `/SSHPASSWORD` und `/SSHKEY` (siehe unten)

`/SSHUSER:username`

Anstatt den Benutzernamen mit dem `/SSH` Parameter anzugeben (siehe oben), kann der Benutzername auch als separater Parameter übergeben werden. Dies ist besonders hilfreich, wenn der Benutzername Sonderzeichen enthält, die mit der `user:password@host` Syntax in Konflikt geraten könnten.

`/SSHPASSWORD:password`

Anstatt das Passwort mit dem `/SSH` Parameter anzugeben (siehe oben), kann das Passwort auch als separater Parameter übergeben werden. Dies ist besonders hilfreich, wenn das Passwort Sonderzeichen enthält, die mit der `/SSH:user:password@host` Syntax in Konflikt geraten würden.

`/SSHKEY:datei`

In Verbindung mit dem `/SSH` Parameter kann, wenn im SSH-Parameter ein Benutzer angegeben ist, mit Hilfe des `/SSHKEY` Parameters zusätzlich zum Kennwort (oder statt Kennwort) eine Schlüsseldatei zu Authentifizierung mit dem Public/Private Key Verfahren angegeben werden. Die Angabe ist hierbei entweder ein Dateiname aus dem ZOC SSH Verzeichnis oder eine Angabe mit voll qualifiziertem Pfad:

```
ZOC /SSH:harry@secure.hogwarts.edu /SSHKEY:id_dsa
```

Alternativ kann die Schlüsseldatei auch im `/SSH` Parameter mit einem weiteren Doppelpunkt hinter dem Kennwort angegeben werden, z.B.

```
ZOC /SSH:harry:alohomora:id_dsa@secure.hogwarts.edu oder  
ZOC /SSH:harry::id_dsa@secure.hogwarts.edu
```

/TELNET:ziel

Bei angegebener Option baut ZOC direkt nach dem Programmstart eine Telnet-Verbindung zum angegebenen Host auf, z.B. `ZOC /TELNET:bbs.channell1.com`

```
ZOC /TELNET:192.168.0.5:110
```

```
ZOC -telnet secure.hogwarts.edu smtp.
```

/TT

Diese Option bewirkt, dass ZOC beim Laden eine andere Zeichenumsetzungstabelle lädt, z.B.

```
ZOC /TT:SCAND.ZTR.
```

/U

Diese Option steht für 'unbeaufsichtigt'. Während des Programmablaufs erscheinen dann keine Abfragefenster (wie z.B. 'Sind Sie sicher'), sondern es werden die Standard-Ergebnisse der Fenster ausgeführt. So können automatische Abläufe gestartet werden, ohne dass wegen einer Abfrage die gesamte Verarbeitung unterbrochen wird.

/WINPOS:x,y

Der `/WINPOS` Parameter erlaubt es, die Position und Größe des Fensters beim Starten zu kontrollieren. `x/y` sind Pixelkoordinaten von der linken, oberen Ecke gemessen, Breite/Höhe steuert die Fensterbreite in Bildpunkten. Die Breite und Höhe kann von ZOC während dem Starten entsprechend der Layout-Optionen des Verbindungsprofils verändert werden, z.B. den um die Größe des Terminals mit der dort eingestellten Schriftart darzustellen.

/WD:ordner

Mit dieser Option kann das Arbeitsverzeichnis angegeben werden, aus dem Datendateien geladen werden (Optionen, Skripts, usw.). Siehe auch [Order für Benutzerdaten](#).

1.2.2 Kommandozeilendateien (commandline.ini, *.zsh/.zln)

COMMANDLINE.INI

Die Vielzahl der Kommandozeilenparameter kann aus dem Starten von ZOC eine etwas unübersichtliche Angelegenheit machen. Daher können beliebige Parameter, die immer eingestellt sein sollen, auch mit Hilfe einer Datei namens `commandline.ini` an ZOC übergeben werden.

Unter Windows wird diese Datei im ZOC Programmordner gespeichert (im gleichen Ordner wo ZOC.EXE liegt).

Unter macOS erzeugen Sie eine Datei namens `.zoc9_commandline` in Ihrem Benutzerverzeichnis, z.B. `/Users/Name/.zoc9_commandline`

Mit einem Editor wird hierbei [pro Zeile eine Kommandozeilen-Option](#) angegeben.

commandline.ini:

```
/RUN:mystart.zrx  
/NUMLOCK:1
```

ZSH/ZLN Dateien

Falls Sie mit unterschiedlichen Sätzen von Kommandozeilenoptionen arbeiten wollen, können Sie Dateien mit der Endung `.zsh` oder `.zln` erstellen.

Diese Dateiendung ist automatisch mit ZOC verknüpft. Ein Doppelklick auf eine solche Datei startet ZOC und arbeitet die darin enthaltenen Kommandozeilenoptionen ab, als wären Sie im Icon oder der `commandline.ini` angegeben.

Hinweis: Unter macOS ist die `.zsh` Endung evtl. mit einem anderen Programm verknüpft, von daher sollte dort besser die Dateiendung `.zln` verwendet werden.

Server7.zln:

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

1.2.3 ADMIN.INI, Arbeitsverzeichnis, Netzwerk

Arbeitsverzeichnis beim Start

Wenn ZOC von einer Eingabeaufforderung oder von einem Icon aus gestartet wird, wird das dortige Arbeitsverzeichnis ignoriert. Alle Programmdateien werden stattdessen aus dem Verzeichnis geladen, in dem ZOC installiert ist oder aus dem ZOC Datenordner.

Optionsdateien werden aus dann dem Benutzerdatenverzeichnis geladen, das in der Datei `admin.ini` angegeben ist (siehe auch [Information für System/Network Administratoren](#) und [Festlegung des Benutzerdatenordners](#)) oder über den `/WD:` Kommandozeilenparameter.

Falls Sie sich bei Angaben auf der Kommandozeile (z.B. zu startende REXX Skripte) auf das Arbeitsverzeichnis beziehen wollen, können Sie dafür den Platzhalter `%CURDIR%` verwenden, z.B. `/RUN: %CURDIR%\test.zrx`

Die ADMIN.INI Datei

Die Datei `admin.ini` ist im Normalfall der primäre Startpunkt um das Benutzerdatenverzeichnis zu ermitteln, aus dem dann wiederum die weiteren Konfigurationsdateien (Adressbuch, Verbindungsprofile, etc.) geladen werden.

Diese Datei liegt unter Windows im ZOC9 Programmordner und unter macOS innerhalb des `zoc9.app` Pakets im Ordner `Contents`. Eine benutzerdefinierte Version der Datei kann unter macOS als `~/zoc9_admin` abgelegt werden.

Unter beiden Betriebssystemen finden Sie eine Vorlage für diese Datei in Ihrem TEMP Ordner.

Hier werden die Standardnamen für Konfigurationsdateien und auch Einschränkungen für die Funktionen festgelegt, die für den Benutzer zugänglich sind. Über die Datei `admin.ini` bzw. `~/zoc9_admin` ist es auch

möglich, gemeinsame, individuelle oder gemischte Konfigurationen für mehrere Benutzer festzulegen. Im Programmverzeichnis bzw. im TEMP Ordner finden Sie hierfür eine Beispieldatei und die aktuell von ZOC verwendete `admin.ini`, in denen die einzelnen Optionen als Kommentare beschrieben sind. Für den genauen Ablauf wie ZOC die Konfigurationsdateien auswertet, lesen Sie bitte den Text über [Festlegung des Benutzerdatenordners](#).

1.2.4 Festlegung des Benutzerdatenordners

Im Normalfall speichert ZOC alle Konfigurationsdateien in einem zentralen Ordner, der standardmäßig im "Eigene Dateien" Ordner des Benutzers liegt.

Wenn der Ordner beim Start von ZOC nicht existiert (z.B. weil ZOC zum ersten Mal gestartet wurde oder weil der Benutzer den Ordner umbenannt hat), fragt ZOC, ob eine neue Konfiguration erstellt werden oder die Position des Benutzerdatenordners verändert werden soll. Bei Erstellen einer Neukonfiguration werden die Daten aus dem Ordner `newusertemplate` im ZOC Programmverzeichnis in den neu erstellten Ordner umkopiert (Administratoren können im Ordner `newusertemplate` vorkonfigurierte Dateien für die Benutzer hinterlegen).

Die Basis für die Auswahl des Benutzerdatenordners ist die Einstellung `ConfigDataFolder` in der Datei `admin.ini`, die im ZOC Programmverzeichnis liegt. Für Einbenutzerinstallationen legt diese Einstellung immer die Position des Benutzerdatenordners, außer wenn über den `/WD:` Kommandozeilenparameter eine andere Angabe gemacht wurde.

Wenn der Benutzer die Position des Datenordners ändert (z.B. wie oben beschrieben durch Umbenennen), wird die neue Position in die `admin.ini` Datei zurückgeschrieben.

Für Mehrbenutzerinstallationen (d.h. Installationen, bei denen in der `admin.ini` die Einstellung `SharedInstall=yes` eingetragen ist), dient die `admin.ini` zur Festlegung des Standardordners für den Benutzer. Wird der Ordner vom Benutzer verschoben, wird die neue Position nicht in die `admin.ini` eingetragen (da dies alle Benutzer betreffen würde), sondern stattdessen in der Registry unter `HKCU+Software+EmTec+UserConfigFolder` gespeichert.

Das Ändern des Ordners mit Speicherung in der Registry kann in der `admin.ini` über die Einstellung `DisableRegistryConfigFolder=yes` unterbunden werden. (Hinweis: Auch für das Unterbinden der Angabe über den `/WD:` Kommandozeilenparameter existiert dort eine Option.)

Das bedeutet für den Startvorgang von ZOC: Zuerst wird `/WD:` Parameter von der Kommandozeile geprüft. Ist der nicht vorhanden (oder unterbunden) wird in einer Mehrbenutzerinstallation als nächstes der Eintrag `UserConfigFolder` in der Registry geprüft. Ist dieser auch nicht vorhanden (oder deaktiviert), oder handelt es sich um eine Einzelinstallation, legt die Einstellung `ConfigDataFolder` aus der `admin.ini` das Benutzerdatenverzeichnis fest.

Als nächstes wird dann die ZOC Konfigurationsdatei `Standard.zfg` geladen, die sich normalerweise im Benutzerdatenordner befindet, für die aber in der `admin.ini` auch eine andere Position (z.B. in einem gemeinsamen Ordner auf dem Netzwerk) angegeben werden kann. Diese Datei enthält dann die Ordnerangaben für die weiteren ZOC Dateien wie Logs, Skripte, Standardordner für Up- und Download usw.

Dieser Prozess erlaubt ein Maximum an Flexibilität, wenn es darum geht, die Datenstruktur festzulegen. Die Bandbreite reicht hierbei von Einzelinstallationen über Mehrbenutzerinstallation mit individuellen Dateien auf Basis einer Vorgabe vom Administrator (über die Dateien im `newusertemplate`), zu gemischten Installationen (gemeinsame `Standard.zfg` die für die Arbeitsdateien auf Ordner im Bereich der individuellen Nutzer zeigt) bis zu komplett gemeinsamen Daten (gemeinsamer Benutzerdatenordner im Netz).

1.2.5 SET-Variablen

ZOC verwendet die SET-Variable TZ, um das bei Zmodem übertragene Datum der Datei zu korrigieren. Zmodem wurde unter UNIX entwickelt, wo Datum und Uhrzeit immer als 'Greenwich time' gespeichert

sind und die jeweilige lokale Zeit bei jedem DIR-Befehl errechnet wird. Auf diese Weise sind Datum und Uhrzeit einer Datei auf der ganzen Welt immer richtig. Das HPFS-Dateisystem kennt diese Art der Zeitzone-Berechnung, die auch von Zmodem verwendet wird, nicht. Viele Systeme errechnen die Zeitzone auf verschiedene Weise, üblicherweise anhand der 'Eastern Standard Time' (fünf Stunden westlich von Greenwich). Dateien, die von einem solchen Sender empfangen werden, werden von ZOC mit der richtigen Uhrzeit abgespeichert. Leider verwenden einige Gegenstellen andere Zeitzone. Eine Anpassung an eine solche Gegenstelle kann anhand der SET-Variable TZ in der Datei CONFIG.SYS vorgenommen werden.

Die Zeitzone wird anhand SET TZ=<SSS><N><DDD> eingestellt, wobei <SSS> einen 3-stelligen Standard-Zeit-Code, <N> den Zeitabstand von Greenwich und <DDD> einen 3-stelligen Sommer/Winterzeit-Code darstellt, d.h. mit SET TZ=EST5EDT kann z.B. 'Eastern Standard Time' (Standard) oder mit SET TZ=CET2CDT die 'Central European Time' eingestellt werden.

1.2.6 Autostart REXX-Programme

Zwei REXX-Programme werden beim Starten von ZOC automatisch ausgeführt. Das erste Programm, `ZOCEVENT.ZRX` (welches auch beim Schließen von ZOC noch einmal ausgeführt wird), kann dazu verwendet werden, um Fax-Programme (wie z.B. 'FaxWorks') aus- bzw. wieder einzuschalten. Die Datei `ZOCEVENT.TXT` im REXX Ordner enthält hierzu weitere Informationen.

Das zweite Programm heißt `STARTUP.ZRX` und wird erst nach dem Initialisieren von ZOC ausgeführt. Es kann dazu verwendet werden, um z.B. bei jedem Starten von ZOC eine bestimmte Gegenstelle anzuwählen, eine Begrüßung anzuzeigen o.ä. Allerdings wird dieses Skript nur dann abgearbeitet, wenn per Kommandozeilenoption nicht ein anderes Skript gestartet wird und wenn nicht per Kommandozeilenoption eine Verbindung initiiert wird (z.B. `/CALL` oder `/CONNECT` etc).

1.3 Bildschirmelemente

Der ZOC-Bildschirm besteht aus

- ☐ [Menüleiste](#)
- ☐ [Symbolleiste](#)
- ☐ [Benutzerknöpfe](#)
- ☐ [Übertragungsfenster](#)
- ☐ [Eingabezeile](#)
- ☐ [Statuszeile](#)

1.3.1 Die Menüleiste

Von der Menüleiste aus können alle Funktionen des Pulldownmenüs erreicht werden (siehe [Menü-Übersicht](#)).

1.3.2 Die Symbolleiste (Toolbar)

Einige häufig benutzte Funktionen können durch einen Klick auf das entsprechende Klickfeld in der Symbolleiste aufgerufen werden.

Eine Beschreibung für jedes Klickfeld erscheint in der Titelzeile, wenn sich die Maus länger als zwei Sekunden über diesem Klickfeld befindet.

Anhand [Anzeige→Fenster-Elemente→Symbolleiste anpassen](#) oder durch Klicken mit der rechten Maustaste auf eine freie Stelle innerhalb der Symbolleiste kann diese Leiste frei zusammengestellt werden.

1.3.3 Die Benutzerknöpfe

Die Benutzerknöpfe befinden sich unter der Symbolleiste und können als Teil des Verbindungsprofils mit Text oder Aktionen belegt werden.

Die Benutzerknöpfe sind Teil des Verbindungsprofils (Optionen→Verbindungsprofil bearbeiten). ZOC kann unterschiedlichen Verbindungen (Gegenstellen, Hosts) unterschiedliche Verbindungsprofile zuordnen, was bedeutet, dass Sie theoretisch für jede Gegenstelle eine andere Leiste mit Knöpfen verwenden können.

In den Einstellungen der Benutzerknöpfe kann der entsprechende Befehltext für jeden Knopf angegeben werden. Dieser Befehltext wird als Tooltip angezeigt, wenn die Maus länger als zwei Sekunden auf diesem Knopf steht.

1.3.4 Das Übertragungsfenster

Im Übertragungsfenster werden die empfangenen Zeichen angezeigt. Das Aussehen der Zeichen

(Schriftart, Größe, Farbe wird im Wesentlichen in Optionen, Verbindungsprofil, Zeichensatz/Fenster) eingestellt.

In der Standard-Konfiguration öffnet sich mit einem rechten Mausklick aus das Hauptfenster das Pop-up-Menü.

Mit der linken Maustaste kann Text auf dem Bildschirm markiert werden.

Wird während des Markierens die Alt-Taste gedrückt, wird die Markierung blockweise durchgeführt.

Wird während des Markierens die Strg-Taste gedrückt, so wird der Text in der Zwischenablage nicht überschrieben, sondern angehängt.

Der markierte Text wird vom Programm in die Zwischenablage kopiert. Über die Funktionen [Einfügen](#), [Einfügen ohne Zeilenwechsel](#), [Einfügen als Zitat](#), [Einfügen als CIS-Zitat](#), [Zwischenablage Drucken](#) und [Zwischenablage Editieren](#) im [Bearbeiten-Menü](#) und mit einigen Shortcuts kann darauf wieder zugegriffen werden.

1.3.5 Die Eingabezeile

Die Eingabezeile ist ein Eingabefeld, in das Text eingegeben werden kann. Der eingegebene Text wird erst beim nächsten Return gesendet. Diese Funktion ist bei Online-Unterhaltungen sehr hilfreich.

Die Editier-Zeile kann durch [Optionen→Verbindungsprofil→Fenster](#) oder durch die Tastenkombination Alt+C oder durch das Drücken der Scroll-Lock (Rollen) Taste aktiviert werden (letzteres nur, wenn die entsprechende Auswahl in [Optionen→Verbindungsprofil→Terminal](#) getroffen ist).

Wenn der Cursor in der Eingabezeile steht, kann durch Drücken der Tastenkombination Alt+C oder der Scroll-Lock (Rollen) Taste die Eingabezeile ausgeschaltet werden. Mit aktiver Eingabezeile kann mit der Esc oder Tabulatortaste der Cursor zurück ins Hauptfenster gebracht werden.

Durch Drücken der Pfeiltasten nach oben und unten kann auf die letzten Eingaben wieder zugegriffen werden (ähnlich der History-Funktion in einer Windows-Befehlszeile).

1.3.6 Die Statuszeile

Die Statuszeile zeigt einige Informationen und bietet Zugriff auf die entsprechenden Optionen. (Bei Verwendung eines sehr kleinen Zeichensatzes kann das ZOC-Fenster so klein sein, dass einige Informationen nicht sichtbar sind).

Wenn sich die Maus länger als zwei Sekunden über einem Element der Statuszeile befindet (Felder und LEDs), wird eine kleine Beschreibung als Tooltip angezeigt.

Verbindungstyp Parameter

Dieses Feld zeigt die momentan eingestellte Verbindungsmethode (Verbindungstyp) und bei serieller Kommunikation die Baudrate sowie die Übertragungsparameter an. Durch Anklicken wird die Dialogseite [Optionen→Verbindungsprofil→Verbindungstyp](#) angezeigt, in der alle Verbindungsmethoden ausgewählt und konfiguriert werden können.

Emulation

Dieses Feld zeigt die momentan aktive Bildschirmemulation an. Durch Anklicken wird die Dialogseite [Optionen→Verbindungsprofil→Emulation](#) angezeigt, in der aus einer Liste verfügbarer Emulationen ausgewählt werden kann.

Transfer Protokoll

Dieses Feld zeigt das momentan ausgewählte Übertragungsprotokoll an. Durch Anklicken wird die Dialogseite [Optionen→Verbindungsprofil→Dateitransfer](#) angezeigt, in der alle Transferprotokolle ausgewählt und konfiguriert werden können.

LEDs

Die vier LEDs in der Statuszeile bieten zwei Funktionalitäten.

In grüner Farbe stellen die 4 LEDs die originalen VT102-LEDs dar und werden per VT102-

Steuersequenzen gesteuert. Die LEDs bleiben grün, bis entweder die Emulation gewechselt oder die Funktion Bildschirm rücksetzen/löschen ausgewählt wird. Diese beiden LEDs können in [Optionen→Verbindungsprofil→Fenster](#) abgeschaltet werden. In roter Farbe zeigen sie zusätzliche Statusinformationen an.

LED 1

Hellrot: Skript-Abarbeitung aktiv.

LED 2

Blau: Tastenmodus Doorway (DOS Tasten) aktiv, Pink: AutoLogin aktiv, Grün: REXX oder AutoLogin lernen.

LED 3/4

Empfangs/Sende-LEDs.

Logdatei

Diese Anzeige enthält den Namen der aktuellen Logdatei und zeigt an, ob die Protokollierung ein- oder ausgeschaltet ist.

Online-Zeit

Diese Anzeige enthält die aktuelle Verbindungszeit.

Fenstergröße

Zeigt die Größe des Übertragungsfensters (Spalte mal Zeile). bzw. Cursorposition an (einstellbar in Optionen, Programmeinstellungen, Fenster).

1.4 Menü-Befehle

Das Hauptmenü besteht aus einigen Untermenüs, mit denen auf die Funktionen von ZOC zugegriffen werden kann:

- ☐ [Datei-Menü](#)
- ☐ [Bearbeiten-Menü](#)
- ☐ [Anzeige-Menü](#)
- ☐ [Log-Menü](#)
- ☐ [Transfer-Menü](#)
- ☐ [Skript-Menü](#)
- ☐ [Extras-Menü](#)
- ☐ [Optionen-Menü](#)

1.4.1 Datei-Menü

Das Datei-Menü enthält alle notwendigen Funktionen, um die Kommunikationsgeräte (Verbindungstypen, z.B. Modem, Telnet, ISDN, SSH) zu steuern und Verbindungen zu Gegenstellen herzustellen. Außerdem können einige entsprechend gekennzeichnete Adressbucheinträge direkt aus diesem Menü heraus ausgewählt werden.

Einige Verbindungsmethoden bieten zusätzliche Funktionen an, die in diesem Menü erscheinen (wie z.B. *Are you there* für Telnet). Auch diese Funktionen sind in den jeweiligen Verbindungstyp-Hilfen beschrieben.

Neue Verbindung

Diese Funktion öffnet einen Dialog, um sich mit einem Host unter Verwendung aller notwendiger Einstellungen zu verbinden, ohne einen Eintrag im Adressbuch vornehmen zu können (siehe [Neue Verbindung](#)).

Adressbuch

Das Adressbuch bietet die Möglichkeit, beliebig viele Gegenstellen zu speichern, und eine oder mehrere Gegenstellen anzuwählen (siehe [Adressbuch](#)).

Wieder verbinden

Falls im Fenster schon einmal eine Verbindung bestand, wird eine neue Verbindung zum gleichen Rechner aufgebaut.

Verbindung trennen

Unterbricht die momentane Verbindung. Bei Verwendung eines Modems wird das Auflege-Kommando aus [Einstellungen Modem](#) verwendet.

Neues Fenster

Startet eine neue ZOC-Instanz. Auf diese Weise sind mehrere gleichzeitige Verbindungen möglich.

Neues Registerblatt

Öffnet ein neues Registerblatt im aktuellen Fenster.

Fenster schließen

Schließt das aktuelle Fenster.

Bildschirm drucken

Diese Funktion druckt den Inhalt des Terminalfensters auf den unter Optionen, Programmeinstellungen, Drucker ausgewählten Drucker.

Bildschirm speichern

Speichert den Inhalt des Terminalfensters in eine unter [Optionen→Programmeinstellungen→Spez. Dateien](#) angegebene Datei.

Verbindungen annehmen

Diese Funktion aktiviert die Verbindungsannahme des gerade aktiven Verbindungstyps. Der Verbindungstyp Telnet erzeugt einen eingehenden Port 23. Serial/Modem als Verbindungstyp schaltet den Auto-Answer Modus des Modems anhand der in den Modemeinstellungen angegebenen Answer-Kommandos ein oder aus. Im Auto-Answer Modus nimmt das Modem ankommende Anrufe automatisch entgegen und versucht, eine Verbindung zum anrufenden Modem herzustellen (was natürlich nicht sinnvoll ist, wenn auch normale Anrufe auf dieser Leitung erwartet werden).

Nächsten Eintrag anwählen

Wird im Adressbuch mehr als ein Eintrag ausgewählt, so kann mit dieser Funktion der nächste Host angewählt werden, nachdem die erste Verbindung zustande gekommen ist.

AutoLogin abbrechen

Diese Funktion bricht den automatischen Login eines Adressbucheintrages ab (siehe [Adressbucheinträge ändern](#)). AutoLogins werden durch das pinkfarbene Leuchten der zweiten LED in der Statuszeile angezeigt.

Verbindung zurücksetzen

Diese Funktion setzt das gerade aktive Verbindungsgerät in seinen Grundzustand zurück. Einem Modem wird das unter [Einstellungen Modem](#) angegebene Init-Kommando gesendet. Diese Funktion wird unabhängig vom aktuellen Carrier-Detect Status immer ausgeführt.

Break senden

Diese Funktion sendet für eine definierte Zeit ein Break-Signal, d.h. zieht die Datenleitung des Modems auf low (siehe [Einstellungen Modem](#)). Einige Gegenstellen benutzen dieses Break, um die aktuelle Verarbeitung abubrechen. Bei anderen eingestellten Kommunikationsmethoden wie Telnet wird eine ähnliche Funktion ausgeführt. Bei Methoden die nichts Vergleichbares bieten, wird die Funktion ignoriert.

Funktionen abhängig vom Verbindungstyp

Funktionen abhängig vom Verbindungstyp (z.B. Telnet) sind im Hilfesystem unter [Verbindungsmethoden](#) beschrieben.

Verbindung zu XXXX

Am Ende des Datei-Menüs stehen alle Adressbucheinträge, deren Menüzugriffs-Option im Adressbuch eingeschaltet ist, zum direkten Auswählen bereit (siehe [Adressbucheinträge ändern](#)). Dies ist gerade für oft benutzte Einträge sehr sinnvoll.

1.4.2 Bearbeiten-Menü

Das Bearbeiten-Menü verwaltet den Text, der in der Zwischenablage des Betriebssystems gespeichert ist. Normalerweise wird Text durch Markieren mit der linken Maustaste im ZOC-Fenster in die Zwischenablage kopiert (kann mit der Alt-Taste kombiniert werden), es können aber auch Texte aus Editoren und beliebigen anderen Programmen in die Zwischenablage gestellt werden.

Siehe auch: Optionen, Programmeinstellungen, [Zwischenablage](#)

Kopieren

Mit dieser Funktion wird der am Bildschirm markierte Text in die Zwischenablage kopiert.

Fensterinhalt kopieren

Diese Funktion kopiert den Inhalt des gesamten ZOC-Fensters in die Zwischenablage.

Zurückblätternpuffer kopieren

Kopiert den gesamten Inhalt des Puffers zum Zurückblättern in die Zwischenablage.

Einfügen

Sendet die in der Zwischenablage befindlichen Zeichen an die Gegenstelle. Da der Text ähnlich wie bei einem ASCII-Upload übertragen wird, gilt die in [Optionen→Verbindungsprofil→Text Senden](#) eingestellte Verzögerung der Übertragung und die CR/LF-Übersetzung auch für diese Funktion.

Einfügen ohne Zeilenwechsel

Sendet die sich im Zwischenspeicher befindenden Zeichen an die Gegenstelle. Ein Zeilenvorschub wird gegen ein Leerzeichen ersetzt. Die Geschwindigkeit der Übertragung kann in [Optionen→Verbindungsprofil→Text Senden](#) eingestellt werden.

Eine Anwendung dieser Funktion ist das Auswählen von Dateinamen aus einer gesendeten Liste. Nach Übertragung der Liste mit allen Dateinamen können blockweise mehrere Dateinamen markiert (linke Maustaste zusammen mit der Alt-Taste) und mit dieser Funktion hintereinander einem Download-Befehl der Gegenstelle angehängt werden.

Einfügen fortsetzen

Dieser Befehl ist während TN3270 Verbindungen verfügbar, wenn bei vorherigen Einfügen der Text nicht in die Bildschirmfelder passte und das Einfügen deshalb abgebrochen wurde.

Zwischenablage an Editor

Diese Funktion startet den in [Optionen→Programmeinstellungen→Spez. Dateien](#) angegebenen Editor mit dem Inhalt der Zwischenablage.

Zwischenablage drucken

Sendet den Inhalt der Zwischenablage an den Drucker. Der Drucker muss in [Optionen→Programmeinstellungen→Drucker](#) angegeben sein.

1.4.3 Anzeige-Menü

ZOC verfügt über verschiedene Protokollierungsmethoden um Daten später wieder anzeigen zu können. Alle empfangenen Zeichen seit Programmstart werden im Hauptspeicher mitprotokolliert damit auch Text, der nicht mehr am Bildschirm steht, noch gelesen und verwendet werden kann. Hierfür stehen der Puffer zum Zurückblättern (der die Daten aufnimmt, die oben aus dem Bildschirm "herausrollen") und das sogenannte Datenprotokollfenster (das die Daten in der Reihenfolge protokolliert, in der sie empfangen werden) zur Verfügung.

Siehe auch: [Protokollierung in Datei](#)

Zurückblättern

Diese Funktion zeigt den Inhalt des Anzeige-puffers im ZOC Fenster an und kann auch aktiviert werden, indem der Rollbalken im ZOC Fenster bedient wird. Einstellungen hierzu finden sich in [Optionen→Programmeinstellungen→Zurückblättern](#)

Zurückblättern als Fenster

Alternativ zum Zurückblättern im Hauptfenster kann das Zurückblättern auch in einem eigenen Fenster erfolgen, so dass parallel der alte und der aktuelle Text sichtbar sind. Die Schriftart für dieses Fenster wird in Optionen, Programmeinstellungen, [Zurückblättern](#) eingestellt.

Löschen Zurückblättern

Löscht den Anzeigepuffer. Eine Sicherheitsabfrage vor dem Löschen kann in [Optionen→Programmeinstellungen→Abfragen](#) aktiviert werden.

Suchen

Funktion zum Suchen von Text innerhalb des Puffers zum Zurückblättern.

Zurückblättern in Editor

Speichert den gesamten Inhalt des Puffers zum Zurückblättern in Datei und öffnet danach die Datei im Editor.

Datenprotokollfenster anzeigen

Zeigt das Fenster des Datenprotokoll-Puffers an. Der Datenprotokoll-Puffer ist eine alternative Form des Puffers zum Zurückblättern. Er speichert die Daten in der Reihenfolge wie sie empfangen werden, bzw. kann auch [Verbindungsprofil→Trace/Debug](#) Daten zeigen.

Datenprotokollpuffer löschen

Löscht den Inhalt des Datenprotokollfensters (siehe oben).

Nächstes/Vorheriges Registerblatt

Wechselt zum vorherigen bzw. nächsten Registerblatt innerhalb eines Fensters.

Miniaturvorschau

Zeigt die Sitzungen aller Registerblätter in miniaturisierter Form an.

Fenster-Elemente

Öffnet ein Untermenü, mit dem alle Fensterelemente von ZOC ein- und ausgeschaltet werden können. Dieses Menü hat die gleiche Funktion wie die entsprechenden Optionen auf der Fensterseite der [Verbindungsprofile](#).

Schnipsel

ZOC überprüft im Hintergrund alle ankommenden Daten auf Dateinamen und Internet-Adressen und stellt diese in einem kleinen Fenster zusammen. Mit dieser Funktion kann das Schnipsel-Fenster angezeigt oder verdeckt werden. Bei angezeigtem Fenster kann durch Doppelklick auf einen entsprechenden Text dieser Text gesendet werden.

Bildschirm löschen/rücksetzen

Diese Funktion löscht das Terminalfenster und setzt die LEDs in der [Statuszeile](#) auf rot zurück.

Editor

Ruft den in [Optionen→Programmeinstellungen→Spez. Dateien](#) angegebenen Editor auf.

Eingabeaufforderung

Ruft die in [Optionen→Programmeinstellungen→Spez. Dateien](#) angegebene Shell auf.

Chat Modus

Diese Funktion teilt den Bildschirm horizontal in zwei Teile und zeigt den selbst eingegebenen Text in der oberen Hälfte, den empfangenen Text in der unteren Hälfte des Bildschirms. Dies ist sehr nützlich, wenn zwei Personen sich über eine Modemverbindung miteinander unterhalten.

Hinweis: Diese Funktion sollte nicht verwendet werden, wenn man mit normalen Gegenstellen, z.B. Mailboxen kommuniziert.

Eingabezeile

Die Eingabezeile kann dazu dienen Text im Voraus zu tippen und erst abzuschicken, wenn Return gedrückt wird (das Versenden geschieht entsprechend der Transfer-2 Ascii Optionen in den

Einstellungen). Mit Alt+C kann zwischen dem Hauptfenster und der Eingabezeile gewechselt werden, mit ESC wird die Eingabezeile wieder ausgeblendet. Wenn mehrere Registerblätter vorhanden sind, ändert Alt+A den Status der "An alle Tabs senden" Option.

1.4.4 Log-Menü

Zusätzlich zu den Funktionen im [Anzeigen Menü](#) (Zurückblättern bzw. Datenprotokollfenster) können alle ankommenden Zeichen permanent in eine Datei auf der Festplatte geschrieben werden (Logdatei). Außerdem können alle empfangenen Zeichen direkt an den Drucker gesendet werden. Die letzte Logfunktion, die Anrufprotokollierung, speichert Zeit, Dauer und Dateiübertragungen aller Verbindungen in einer Datei (siehe [Optionen→Programmeinstellungen→Spez. Dateien](#)).

Das Log-Menü steuert den Puffer, die Logdatei und die direkte Ausgabe auf den Drucker.

Log in Datei

Diese Funktion öffnet und schließt die aktuelle Logdatei. Dies kann auch durch Anklicken des Logdateinamens in der [Statuszeile](#) erfolgen und kann von der Gegenstelle gesteuert werden, wenn die DC2/DC4 Unterstützung eingeschaltet ist (siehe [Optionen→Verbindungsprofil→Log](#)).

Log auf Drucker

Manchmal ist es sinnvoll, Teile einer Übertragung gleichzeitig auf dem Drucker auszugeben. Dies ist möglich, wenn vor der Übertragung der Daten diese Funktion eingeschaltet wird (z.B. vor dem Lesen einer Nachricht).

Zusätzlich ist es möglich, bereits empfangenen Text im Anzeigepuffer-Fenster zu markieren (siehe [Log-Menü](#)) und den Inhalt der Zwischenablage dann im [Bearbeiten-Menü](#) auszudrucken.

Logdatei Name festlegen

Hier kann der Name der Logdatei angegeben werden. Der Name kann auch in jedem Adressbucheintrag angegeben werden. Beim Starten des Programms wird der Standardname aus [Optionen→Verbindungsprofil→Log](#) verwendet. Für den Dateinamen können spezielle Platzhalter verwendet werden, die im Anhang unter [Steuerzeichen](#) beschrieben sind.

Standardname

Beim Anwählen dieses Menüpunktes wird die in [Optionen→Verbindungsprofil→Log](#) angegebene Datei zur aktuellen Logdatei.

Logdateien löschen

Durch die Verwendung von Platzhaltern im Logdateinamen können sehr schnell sehr viele Logdateien entstehen. Mit dieser Funktion können nicht mehr benötigte Logdateien wieder gelöscht werden. Diese Funktion öffnet ein Dateiauswahlfenster, in dem eine oder mehrere Dateien zum Löschen ausgewählt werden können.

1.4.5 Transfer-Menü

Das Transfer-Menü erlaubt das Senden und Empfangen von Dateien. Die Funktionen in diesem Menü werden von den Einstellungen unter [Optionen→Verbindungsprofil→Dateitransfer](#) und [Optionen→Programmeinstellungen→Dateiablage](#) beeinflusst.

Verwandte Themen:

- ☐ [Zmodem Transfer](#)
- ☐ [SCP Transfer](#)
- ☐ [IND\\$FILE Transfer](#)
- ☐ [Secure Shell/SSH](#)
- ☐ [SSH File Transfer](#)
- ☐ [File Transfer Fenster](#)

Upload

Upload nennt man das Senden einer Datei vom eigenen Rechner zur Gegenstelle. Hierfür können von Sender und Empfänger verschiedene Methoden, sog. Übertragungsprotokolle, verwendet werden. Die Uploadfunktion überträgt eine oder mehrere Dateien zur Gegenstelle. Hierbei wird das gerade aktive Protokoll verwendet (siehe [Optionen→Verbindungsprofil→Dateitransfer](#)).

Download

Der Empfang einer Datei von der Gegenstelle heißt Download. Es werden die gleichen Übertragungsprotokolle wie beim Upload verwendet. Die entsprechenden Einstellungen können in [Optionen→Verbindungsprofil→Dateitransfer](#) vorgenommen werden.

Sende Textdatei

Diese Funktion liest Text aus einer Datei und sendet diesen zur Gegenstelle, ohne ein Übertragungsprotokoll zu verwenden. Ein solches Senden ist also nichts anderes als sehr schnell eingetippter Text. Auf diese Weise können z.B. Nachrichten im Editor erstellt und nachträglich gesendet werden.

Hinweis: Sollten bei der Übertragung Zeichen verlorengehen oder die Übertragung zu langsam sein, kann eine Sendeverzögerung für jedes Zeichen in [Optionen→Verbindungsprofil→Text Senden](#) angegeben werden.

Sende Binärdatei

Das Senden einer Binärdatei erfolgt im Wesentlichen so wie das Senden einer Textdatei (s.o.). Jedoch erfolgt keinerlei Umsetzung der Zeichen (z.B. keine Umsetzung von Zeilenende), d.h. die Zeichen werden 1:1 an die Gegenstelle geschickt. Die für Textversand eingestellte Zeichenverzögerung (siehe [Optionen→Verbindungsprofil→Text Senden](#)) wird berücksichtigt (jedoch nicht die Zeilenverzögerung.)

FTP Verbindung

Wenn Sie mit einem SSH- oder Telnet-Server verbunden sind, diese Verbindung durch das Adressbuch zustande gekommen ist, und dort eine FTP-Verbindung für diesen Eintrag hinterlegt ist, dann können Sie diesen Menüpunkt verwenden, um das FTP-Fenster zu öffnen. Dieses Fenster erlaubt Ihnen, auf lokaler und entfernter Seite durch Verzeichnisse zu navigieren und Dateien oder Gruppen von Dateien einfach per Drag-and-Drop zwischen dem lokalen und dem entfernten Computer auszutauschen.

1.4.6 Extras Menü

Dieses Menü enthält Hilfsmittel, die nicht direkt mit der aktuellen Verbindung in Bezug stehen, sondern bei denen es sich eher um Verwaltungsfunktionen allgemeiner Natur handelt, z.B. Funktionen zum Verwalten von SSH-Schlüsseln oder zur Verwaltung und Löschung gespeicherter Kennwörter.

1.4.7 Skript-Menü

REXX Skript starten

Startet ein REXX-Skript (Programm). REXX ist eine Art Batch-Sprache, mit der in ZOC verschiedenste Aufgaben automatisch erledigt werden können (siehe [Die Programmierung von ZOC](#)).

REXX Skript abbrechen

Hält ein gestartetes REXX-Skript an.

REXX Skript bearbeiten

Ruft den eingestellten Editor auf, um ein REXX-Programm zu bearbeiten.

REXX Skript verschlüsseln

Macht ein REXX Skript unlesbar, wobei es für ZOC ausführbar bleibt. Dies kann nützlich sein, um Skripte an Kunden weiterzugeben. Bitte beachten Sie, dass es sich hier um eine relativ einfache Verschlüsselung handelt, die einem talentierten Hacker kaum standhalten wird.

REXX Skript aufzeichnen

Eine der Hauptanwendungen von REXX-Skripten in ZOC ist der automatische Login. Die Erstellung solcher Skripte ist eine eher langweilige und lästige Tätigkeit und kann in den meisten Fällen automatisch erledigt werden. Wenn diese Funktion vor der Anwahl eines Hosts aufgerufen wird, protokolliert ZOC die Loginprozedur im Hintergrund mit und erstellt ein REXX-Skript, das diese Prozedur automatisch abarbeitet.

Hinweis: Das REXX-Skript enthält nur den für das Einloggen notwendigen Code. Es enthält keine Befehle zur Anwahl der Gegenstelle, da die Anwahl im Normalfall vom Adressbuch und nicht vom REXX-Skript erledigt wird.

Aufzeichnung beenden

Nach eingeschalteter [REXX Skript aufzeichnen](#) Funktion und erfolgreichem Einloggen bei einer Gegenstelle wird mit dieser Funktion der Lernprozess beendet. ZOC fragt nach einem Dateinamen und speichert das REXX-Skript entsprechend ab. Dieses Programm kann später in einem Adressbucheintrag angegeben werden (siehe [Adressbucheinträge ändern](#)). ZOC arbeitet dieses Programm dann nach dem Herstellen der Verbindung automatisch ab.

1.4.8 Optionen Überblick

In den Optionen-Menüs können alle ZOC-Parameter verändert werden. Es gibt es die folgenden Arten von Einstellungen:

Programmeinstellungen

Einstellungen, die für alle Gegenstellen gelten.

Verbindungsprofil

Einstellungen die für unterschiedliche Gegenstellen verschieden sein können, z.B. Emulation, Terminalgröße, F-Tasten usw.

Tastatur/Zeichenumsetzungsprofile

Profile die unabhängig vom Verbindungsprofil Einstellungen geladen bzw. Gegenstellen zugeordnet werden können (z.B. Tastenbelegungen für alle VT220 Hosts).

Das Verbindungsprofil und Tastatur- bzw. Zeichenumsetzungsprofile werden in Dateien gespeichert,

wobei jeweils die Datei mit dem Namen `Standard` beim Start von ZOC automatisch geladen wird.

Siehe auch: [Optionen Menü](#), [ZOC konfigurieren](#), [Verbindungsprofile](#) und [Programmeinstellungen](#).

1.4.9 Optionen-Menü

Siehe auch: [Optionen Überblick](#)

Programmeinstellungen

Zeigt den Dialog zur Bearbeitung der sitzungsunabhängigen [Programmeinstellungen](#)

Verbindungsprofil bearbeiten

Öffnet einen Dialog, in dem alle sitzungsbezogenen Einstellungen vorgenommen werden können (siehe [Verbindungsprofile](#)).

Direkt zu

Öffnet ein Untermenü, mit dem auf die entsprechenden Dialogseiten direkt zugegriffen werden kann.

Lade Verbindungsprofil

Lädt ein Verbindungsprofil und nimmt alle Einstellungen vor.

Speichern Verbindungsprofil

Speichert die momentanen Einstellungen in dem momentan geladenen Verbindungsprofil.

Speichern unter

Speichert die momentanen Einstellungen in einem Verbindungsprofil mit neuem Namen. Wird als Dateiname `STANDARD.ZOC` verwendet, werden diese Einstellungen beim Starten von ZOC automatisch vorgenommen.

Verbindungsprofil zurücksetzen

Setzt alle Einstellungen des Verbindungsprofil auf Standardwerte.

Tastaturbelegungsprofile

Zeigt den [Tastaturbelegungsdialog](#) mit dem Sie Dateien mit Tastenbelegungen erstellen und bearbeiten können.

Zeichenumsetzungstabellen

Zeigt den [Zeichenumsetzungstabellen](#) mit dem Sie Dateien mit Zeichenumsetzungstabellen erstellen und bearbeiten können.

1.4.10 Neue Verbindung

Note: Wir bieten verschiedene [Einstiegsanleitungen](#) an, die detaillierte Beschreibungen dieses Dialogs enthalten und sich auf spezifische Kommunikationsmethoden wie [SSH \(Secure Shell\)](#), [Telnet](#) und 10224,serielle Verbindungen) konzentrieren. Im Gegensatz dazu bietet der folgende Text eine allgemeinere Übersicht.

Der Verbindungsdialog ist ein Fenster, das es Ihnen ermöglicht, schnell eine Verbindung zu einem Host herzustellen. Diese Funktion ist für Verbindungen gedacht, die Sie nicht regelmäßig herstellen möchten. Für Hosts, zu denen Sie häufig eine Verbindung herstellen, sollten Sie einen Eintrag im [Adressbuch](#) hinzufügen. Adressbucheinträge können über verschiedene Schnellzugriffsmethoden aktiviert werden, wie

z.B. durch [Benutzerknöpfe](#) oder Tastenkombinationen, und sie bieten zusätzliche Vorteile wie einen [automatischen Login](#).

Um eine Verbindung herzustellen, wählen Sie einen Verbindungstyp, den Host, zu dem Sie eine Verbindung herstellen möchten, eine optionale Portnummer (für Telnet-, Rlogin- und SSH-Verbindungen), eine Emulation und Ihre Anmeldedaten (für SSH- und Rlogin-Verbindungen).

Die Verbindung basiert auf verschiedenen Konfigurationsoptionen (z.B. Fenstergröße, Farben, Cursorform usw.), die in einem [Verbindungsprofil](#) definiert sind. Das Verbindungsprofil enthält auch Optionen für Verbindungstyp und Terminalemulation, aber im Verbindungsfenster können Sie diese spezifischen Einstellungen mithilfe der [Konfigurieren](#) Schaltflächen überschreiben, ohne das Verbindungsprofil zu ändern (eine Änderung des Verbindungsprofils könnte andere Verbindungen beeinflussen, die darauf basieren).

Verbinden mit

Je nach Kommunikationsmethode (Verbindungstyp) wird hier z.B. ein Internet-Hostname, eine IP-Adresse oder eine Telefonnummer angegeben. Dieses Feld hat auch eine Auswahl von Rechnern zum Herunterklappen, mit denen kürzlich eine Verbindung aufgebaut wurde. Falls Sie sich wieder mit einem Rechner verbinden wollen, können Sie ihn einfach aus der Liste auswählen.

Kommunikationsspezifische Hinweise zum Feld 'Verbinden mit'

Local Shell

Für Local Shell geben Sie als Ziel immer [localhost](#) oder [127.0.0.1](#) ein. Alternativ wählen Sie eine Shell im sog. Shebang-Format, z.B. [#!/bin/bash](#) oder [C:\Windows\cmd.exe](#)

Telnet, Rlogin oder SSH

Für Verbindungen per Telnet, SSH oder Rlogin (siehe unten) kann ein Rechnernamen oder IP-Adresse und zusätzlich eine Portnummer oder Bezeichnung eingegeben werden, z.B. [110](#) oder [pop3](#) für Verbindung mit einem POP3 Server. Ohne Portangabe wird automatisch der jeweilige Standardport für die entsprechende Verbindungsart verwendet.

Serial/Direct

Falls Sie sich mit einem Gerät (Router, Nullmodem, etc.) verbinden wollen, das direkt am seriellen Port hängt, wählen Sie eine der folgenden Alternativen:

1. Im Feld [Verbinden mit](#) tragen Sie den Text [localhost](#) oder [serial](#) ein. Dann klicken Sie neben Serial/Direct auf [Konfigurieren](#), wählen einen Kommunikationsport und stellen die seriellen Parameter ein.
2. Im Feld [Verbinden mit](#) tragen Sie den Namen des Kommunikationsports ein (z.B. [COM1](#) unter Windows oder [/dev/cu.pl2303serial](#) unter macOS). Dann klicken Sie neben Serial/Direct auf [Konfigurieren](#) und stellen die seriellen Parameter ein, lassen aber das Feld für den Kommunikationsport leer.

Serial/Modem

Bei Verbindungen per Serial/Modem können Sie ein Wahlkommando angeben. Diese entsprechen den AT-Wahlkommandos, die für Serial/Modem konfiguriert sind (klicken Sie den [Konfigurieren](#) Knopf neben Serial/Modem, um die seriellen Parameter und AT-Befehle zu bearbeiten). Dieser AT Befehl wird dann ans angeschlossene Modem gesendet, um die Verbindung zum entfernten Rechner aufzubauen.

Verbindungsprofil

Alle Verbindungen in ZOC basieren auf einem [Verbindungsprofil](#). Verbindungsprofile enthalten eine Kombination von Optionen (z.B. Fenstergröße, Schriftart, Farben, Funktionstasten usw.), die die

Verbindung und alle anderen Verbindungen beeinflussen, die auf diesem Profil basieren. Das Verbindungsprofil umfasst auch Einstellungen für den Verbindungstyp und die Emulation. Sie können jedoch eine oder beide dieser Einstellungen im Verbindungsfenster überschreiben, indem Sie einen anderen Verbindungstyp und eine andere Emulation auswählen. Dies ermöglicht es Ihnen, ein Verbindungsprofil zu verwenden, das im allgemeinen Ihren Anforderungen entspricht (in Bezug auf Schriftart, Bildschirmlayout, Funktionstasten usw.) und es für verschiedene Verbindungen wiederzuverwenden, selbst wenn diese unterschiedliche Verbindungstypen und Emulationen erfordern.

Siehe auch: [ZOC Konfigurieren](#)

Verbindungstyp

Viele dieser Methoden können angepasst werden, indem Sie auf die [Konfigurieren](#)-Schaltfläche neben der Liste klicken. Diese Einstellungen haben Vorrang vor den gleichen Einstellungen im [Verbindungsprofil](#). Alternativ können Sie die Einstellungen verwenden, die für diesen Verbindungstyp im [Abschnitt Verbindungstyp](#) des Verbindungsprofils konfiguriert sind.

Siehe diesen Link für eine Liste der Verbindungstypen und der dafür verfügbaren Optionen: [Verbindungsprofil→Verbindungstyp](#).

Emulation

Die Emulation ist abhängig von der Gegenstelle und erlaubt dem anderen Rechner, Daten im Terminalbereich von ZOC darzustellen. Bei Verbindungen mit Linux/Unix Hosts, führen wahrscheinlich die Emulationen [Xterm](#) oder [VT220](#) zu guten Ergebnissen. Viele Mailboxen arbeiten dagegen mit der ANSI-BBS Emulation. Emulationsspezifische Einstellungen können mit dem [Konfigurieren](#)-Knopf neben der Emulationsliste vorgenommen werden.

Diese Auswahl überlagert für den Zeitraum der Verbindung evtl. Einstellungen des oben ausgewählten Verbindungsprofils.

Benutzer

Bei Kommunikation über SSH oder Rlogin können Sie einen Benutzernamen angeben, der Sie bei der Gegenstelle identifiziert. Falls Sie hier nichts angeben, werden Sie ggf. nach dem Verbindungsaufbau danach gefragt.

Kennwort

SSH Verbindungen benötigen zusätzlich zum Benutzernamen noch ein Kennwort, das sie zusammen mit dem Benutzernamen identifiziert. Falls das Kennwort nicht angegeben wird, wird es ggf. angefordert.

Wenn Sie die Option [Kennwort speichern](#) verwenden, merkt sich ZOC das Kennwort für den Fall, dass es in diesem Dialog später noch einmal gebraucht wird (gespeicherte Kennwörter können unter [Optionen→Programmeinstellungen→Kennwörter](#) verwaltet werden).

SSH Schlüssel

Für SSH-Verbindungen erfordern bestimmte Gegenstellen Dateien mit öffentlichen und privaten Verschlüsselungsschlüsseln anstelle der Authentifizierung mit Benutzername/Passwort. Sie haben die Möglichkeit, eine individuelle Datei aus dem SSH-Verzeichnis (konfiguriert in [Programmeinstellungen→Dateiablage](#)) auszuwählen, die globalen SSH-Authentifizierungsdateien zu verwenden oder Schlüssel von einem ssh-agent zu nutzen. Um dies zu konfigurieren, klicken Sie auf die [Konfigurieren](#)-Schaltfläche neben der Auswahl [Secure Shell](#) weiter oben in diesem Fenster oder greifen Sie auf die Secure Shell-Einstellungen im [Verbindungsprofil→Verbindungstyp](#)-Abschnitt zu.

Wenn Sie anstelle einer privaten Schlüsseldatei den vollständigen Namen und Pfad einer PKCS#11-kompatiblen DLL oder .dylib angeben, wird ZOC diese pkcs11-DLL verwenden, um auf Ihre Smartcard zuzugreifen und die Identität vom SmartCard-Chip abzurufen.

Hinweis: Öffentliche/private Schlüsselpaar-Dateien können über die Funktion [SSH-Keydateien](#)

erzeugen erstellt werden, die im Extras-Menü von ZOC verfügbar ist.

1.5 Das Adressbuch

Das Adressbuch erreichen Sie über das Datei-Menü und es dient der Speicherung von häufig verwendeten Gegenstellen und deren Adressen und Einstellungen.

Im Hauptfenster können ein oder mehrere Einträge markiert, sowie Funktionen wie Löschen oder Ändern, die für die Bearbeitung des Adressbuches notwendig sind, aufgerufen werden.

Das Adressbuch kann mit Hilfe der [Adressbuch-Einstellungen](#) in mehrere Abschnitte unterteilt werden. Über das Kontextmenü (rechte Maustaste) können pro Abschnitt auch Unterordner angelegt werden.

Verbinden

Diese Funktion verbindet Sie mit einem oder mehreren markierten Einträgen.

Wenn mehr als ein Eintrag ausgewählt ist (zum Markieren müssen die Strg-Taste gedrückt und dann beliebig viele Einträge ausgewählt werden), hängt das Vorgehen von den Einstellungen der einzelnen Einträge ab. Es wird beeinflusst von der Art des Verbindungsaufbaus (AutoVerbindung oder Einfache Verbindung) und ob für den Eintrag ggf. ein neues Fenster oder Registerblatt geöffnet werden soll.

Generell versucht ZOC so viele Verbindungen gleichzeitig aufzubauen wie möglich. Dies wird allerdings von der Tatsache begrenzt, dass pro Fenster nur ein AutoVerbindungs-Dialog angezeigt werden kann, so dass ggf. Einträge in einer Warteschlange abgelegt werden. Um diese dann anzuwählen, muss nach Beenden einer Verbindung wieder Befehl [Nächster Eintrag](#) im [Datei-Menü](#) aufgerufen werden.

Falls Sie keinen Verbindungsaufbau mit Warteschlange wünschen, konfigurieren Sie die Einträge idealerweise mit dem Typ Einfache Verbindung (im Abschnitt [Gegenstelle](#) der Einträge) und geben an, dass ggf. ein neues Registerblatt geöffnet werden soll (im Abschnitt [Fenster](#) der Einträge).

Neu

Erzeugt und editiert einen neuen Eintrag im gerade geöffneten Adressbuch.

Bearbeiten

Ändert den Inhalt eines ausgewählten Eintrages (siehe [Adressbucheinträge ändern](#)).

Kopie

Erzeugt einen neuen Eintrag und kopiert den Inhalt des angewählten Eintrags in den neuen Eintrag.

Löschen

Löscht einen oder mehrere Einträge aus dem Adressbuch.

Verknüpfung erstellen

Erstellt eine Verknüpfung zu einem Adressbucheintrag als Icon auf der Arbeitsoberfläche oder als Feld in der Leiste der ZOC Benutzerknöpfe.

Verschieben

Verschiebt Einträge in einen anderen Abschnitt des Adressbuchs.

Abschnitte verwalten

Hier können beliebige Bezeichnungen für die Abschnitte im Adressbuch angegeben werden bzw. Abschnitte hinzugefügt oder gelöscht werden.

Importieren ...

Importiert Adressbuchdateien von anderen Terminalprogrammen.

Drucken

Druckt alle markierten Einträge aus.

Fällige markieren

Markiert alle Einträge, die als fällig gekennzeichnet sind (diese Einträge haben einen kleinen gelben Blitz vor der Telefonnummer). ZOC erkennt anhand des Datums des letzten Anrufs und dem Feld 'Anruf alle xx Tage', welche Einträge wieder angerufen werden sollen.

Einstellungen

Öffnet die Adressbucheinstellungen zum Einstellen der Spalten, Abschnitte und weitere Optionen. Siehe auch [Adressbuch-Einstellungen](#).

1.5.1 Adressbucheinträge konfigurieren

Ein Eintrag im Adressbuch kombiniert verschiedene Einstellungsprofile, die im Optionen Menü von ZOC (siehe [Optionen Menü](#) und [ZOC konfigurieren](#)) erstellt und gespeichert wurden.

Die Charakteristika einer Verbindung sind eine Kombination aus [Programmeinstellungen](#), [Verbindungsprofil](#), [Tastaturbelegung](#), [Zeichenumsetzung](#), sowie den Einstellungen im Adressbucheintrag für Verbindungstyp, Emulation, Dateiübertragungsprotokoll, Zeichensatz (Codepage), Upload-/Download-Ordner.

Für die Dauer der Verbindung überlagern hierbei die direkt im Eintrag vorgenommenen Einstellungen (z.B. Art der Verbindung, Emulation oder Download-Ordner) diejenigen aus den o.g. dateibasierten Einstellungen. Das heißt, die Angabe der Verbindungseinstellungen im Adressbucheintrag sticht die Verbindungstyp-Angabe im Verbindungsprofil. Der Download-Ordner aus dem Eintrag gewinnt gegenüber dem Download-Ordner in den Programmeinstellungen, usw.

Die einzelnen Einträge bestehen aus

- ☐ [Gegenstelle](#)
- ☐ [Optionen](#)
- ☐ [Login](#)
- ☐ [FTP](#)
- ☐ [Verknüpfungen](#)
- ☐ [Fenster](#)
- ☐ [Mailbox](#)
- ☐ [Info](#)

1.5.1.1 Gegenstelle

Bezeichnung

Benennt den Adressbucheintrag. Diese Bezeichnung wird im Adressbuch angezeigt und dient auch als Titel für das Registerblatt im Hauptfenster wenn eine Verbindung aufgebaut wird.

Abschnitt

Hier muss angegeben werden, in welchem Abschnitt der Eintrag gespeichert werden soll. Die Abschnitte können in [Adressbuch-Einstellungen](#) verwaltet werden. Das Verschieben in andere Abschnitte oder Ordner ist auch im Hauptfenster des Adressbuchs über die rechte Maustaste möglich.

Symbol

Für jeden Eintrag kann ein farbiges Symbol aus der Liste gewählt werden. Dieses Symbol wird für den Eintrag im Adressbuch angezeigt und es wird auch für das ZOC Hauptfenster verwendet wenn eine Verbindung besteht.

Außerdem wird die Farbe in die Registerblätter im Hauptfenster übernommen, wenn ein Registerblatt mit Verbindung für diesen Eintrag erstellt wird.

In der Liste oben einsortieren

Die Einträge im Adressbuch werden anhand der Spalten sortiert, die Sie im Hauptfenster anklicken. Diese Option dient dazu, wichtige Einträge unabhängig von der Sortierreihenfolge nach vorne zu sortieren.

Verbinden mit

Hier wird die Zieladresse der Gegenstelle (z.B. IP-Adresse und Portnummer bei SSH- oder Telnetverbindungen) bzw. die Telefonnummer (bei ISDN oder Modem) angegeben.

Verbindungsaufbau

Für den Verbindungsaufbau stehen die sog. AutoVerbindung (mit Dialog und Wiederholungsfunktion) oder die einfache Verbindung (kein Dialog, nur ein einziger Verbindungsversuch) zur Verfügung.

Letzteres ist sinnvoll für Netzwerkverbindungen (SSH, Telnet, etc.) und erlaubt den Aufbau mehrerer Verbindungen gleichzeitig, falls im Adressbuch mehrere Einträge zum Verbinden ausgewählt sind und falls diese in den [Fenstereinstellungen](#) das Öffnen der Verbindung in einem eigenen Registerblatt konfiguriert haben.

Bei AutoVerbindung werden die Verbindungen immer nacheinander aufgebaut, da ZOC immer nur ein AutoVerbindungsfenster anzeigen kann.

Siehe auch: Funktion [Verbinden](#) in [Adressbuchfenster](#)

Verbindungsprofil

Hier kann ein zu ladendes Verbindungsprofil angegeben werden (z.B. [Standard.zoc](#) oder [TN3270.ZOC](#)). Verbindungsprofile sind Sammlungen von verbindungsrelevanten Optionen, die mehreren ähnlichen Gegenstellen zugeordnet werden können.

Auf diese Weise können für verschiedene Gegenstellen verschiedene Verbindungsprofile, oder für ähnliche Gegenstellen das gleiche Verbindungsprofil verwendet werden (neue Verbindungsprofile können mit dem Knopf [Speichern unter](#) im Verbindungsprofildialog erzeugt werden).

Nach dem Laden des Verbindungsprofils wird die Kommunikationsmethode, die Emulation und das Übertragungsprotokoll sowie der Zeichensatz aus dem Adressbucheintrag gesetzt, d.h. diese Einstellungen aus dem Adressbucheintrag können die Einstellungen die im Verbindungsprofil gemacht sind überlagern. Dadurch ist es möglich, das gleiche Profil (z.B. die dort festgelegten Benutzerknöpfe, Fenstereinstellungen, usw.) zu verwenden obwohl die Einträge im Adressbuch unterschiedliche Kommunikationsmethoden oder Emulationen verwenden.

Siehe auch [Adressbucheinträge konfigurieren](#)

Verbindungstyp

Hier kann eine Kommunikationsmethode (und dazu gehörende Einstellungen) für speziell diesen Adressbucheintrag ausgewählt werden.

Für die meisten Kommunikationsmethoden existieren weitere Optionen, die mit dem [Konfigurieren](#)-Knopf neben der Liste eingestellt werden können. Damit ist es z.B. möglich, für verschiedene SSH-Verbindungen unterschiedliche Authentifizierungsmethoden zu wählen. Die einzelnen Optionen hängen von der jeweils verwendeten Kommunikationsmethode ab.

Die Auswahl der Kommunikationsmethode und deren Einstellungen überlagern ggf. für die Dauer der Verbindung die Verbindungstyp-Einstellungen aus dem Verbindungsprofil.

Siehe auch [Adressbucheinträge konfigurieren](#)

Emulation

Die hier angegebene Emulation wird nach dem Laden des Verbindungsprofils und vor dem Anwählen der Gegenstelle aktiviert. Der [Konfigurieren](#)-Knopf neben der Liste enthält ggf. weitere Optionen, die sich auf die Emulation beziehen.

Die Auswahl der Emulation und deren Einstellungen überlagern ggf. für die Dauer der Verbindung die Emulations-Einstellungen aus dem Verbindungsprofil.

Siehe auch [Adressbucheinträge konfigurieren](#)

1.5.1.2 Optionen

Tastaturbelegungsprofil

Hier kann eine Tastaturliste (kann unter Optionen, Tastatur erstellt werden) angegeben werden, die für diesen Eintrag verwendet wird.

Zeichenumsetzungstabelle

Hier kann eine Zeichenumsetzungstabelle (kann unter Optionen, Tabellen erstellt werden) angegeben werden, die für diesen Eintrag verwendet wird.

Dateitransferprotokoll

Das hier angegebene Übertragungs-Protokoll wird vor dem Anwählen der Gegenstelle ausgewählt und kann die Einstellung im ausgewählten Verbindungsprofil überlagern.

Zeichensatz/Codepage

Die Codepage steuert das Zusammenspiel von Zeichendarstellung und übertragenen Codes und kann die Einstellung im ausgewählten Verbindungsprofil überlagern.

Siehe auch [Optionen→Verbindungsprofil→Schriftart](#)

Logdatei

In die hier angegebene Logdatei werden alle von dieser Verbindung mitprotokollierten Daten gespeichert. Der Name kann optional einen absoluten oder relativen Pfad enthalten (Ordner im Pfad werden bei Bedarf neu erzeugt).

Spezielle Platzhalter für Datum, Uhrzeit usw. können im Namen angegeben werden, wie für den Lognamen in [Optionen→Verbindungsprofil→Log](#) beschrieben. Beispiele sind `^+_^1^2^3.log` oder `Host_+\\Log_^3_^2_^1.log`.

Der anfängliche Logging-Zustand (Aktiv/Inaktiv) wird vom Verbindungsprofil bestimmt.

Wird das Feld leer gelassen, wird der Name aus dem Verbindungsprofil benutzt.

Download/Upload-Ordner

Hier können Verzeichnisse angegeben werden, in die (oder aus denen) Dateien per Datenübertragung gespeichert (oder verschickt) werden. Wenn die Felder ausgefüllt sind, überlagert diese Einstellung die Standardordner für Upload/Download für die Zeit die Sie mit dieser Gegenstelle verbunden sind.

Siehe auch [Optionen→Programmeinstellungen→Dateiablage](#)

1.5.1.3 Login

Benutzer

Hier kann ein Name für den Login angegeben werden. Diese Einstellung wird bei Verwendung von Secure Shell, Rlogin bzw. im AutoLogin (s.u.) verwendet.

Bei Angabe von `%USERNAME%` oder `%USERNAME%LWR%` oder `%USERNAME%UPR%` wird der Name des Benutzers verwendet, der momentan an Ihrem lokalen Computer angemeldet ist (die letzten beiden Platzhalter beinhalten den Benutzernamen umgesetzt in Klein- bzw. Großbuchstaben). Auf Windows-Computern ist auch der Platzhalter `%DOMAINNAME%` verfügbar, der durch die mit der Anmeldung verknüpfte Domäne oder Arbeitsgruppe ersetzt wird.

Falls das Feld ein @-Zeichen und direkt darauf folgend einen Dateinamen enthält, z.B. `@C:\users\you\username.txt` oder `@%ZOCFILES%\username.txt`, dann wird die erste Zeile aus der Datei als Benutzername verwendet.

Außerdem kann der Inhalt des Feldes (wie bei Kennwort beschrieben) mit dem Steuercode `^%` auf Tasten oder Benutzerknöpfe belegt werden.

Kennwort

Hier kann das Kennwort zum Einloggen in die Gegenstelle eingegeben werden (mit Secure Shell oder AutoLogin). Außerdem kann das Kennwort nach Verbindungsaufbau anhand einer Funktionstaste gesendet werden, wenn diese die mit Wert `^&` belegt ist.

Wenn also z.B. die F12-Taste mit dem Text `^&` belegt ist, genügt auf die Frage nach dem Kennwort ein Druck auf F12 (siehe [Optionen→Verbindungsprofil→Funktionstasten](#)).

Die AutoLogin Funktion (siehe unten) ist allerdings eine komfortablere Methode, um den Login zu automatisieren. Diese Funktion greift auch auf das Kennwortfeld zurück, und verwendet den Platzhalter, damit das Kennwort dort nicht im Klartext zu lesen ist.

SSH Schlüssel

Für SSH Verbindungen ist es möglich, die Authentifizierung mit Identitätsdateien (Private Key Files) durchzuführen.

Falls Sie statt einer Private-Key Datei, den vollen Namen und Pfad einer PKCS#11 kompatiblen DLL angeben, wird ZOC die PKCS#11 DLL nutzen, um auf Ihre Smart-Card zuzugreifen und somit Ihre Identität von der SmartCard zu lesen.

Weiterhin ist es möglich, einen Schlüssel aus dem Windows Zertifikats-Speicher auszuwählen (diese beinhalten auch Schlüssel von Windows-kompatibler Hardware, wie z.B. CAC/PIV Leser und die meistverwendeten USB-basierten Schlüssel).

Ebenfalls möglich ist die Verwendung von Standard-Schlüsseldateien für den Login (diese Dateien werden unter [SSH→Globale Schlüsseldateien](#) konfiguriert).

Hinweis: SSH-Authentifizierung ist nur verfügbar, wenn Sie auch einen Benutzernamen angeben.

Login-Skript/Parameter

Eine hier angegebene [REXX](#)-Datei wird nach erfolgreichem Verbindungsaufbau ausgeführt und der optionale Parameter ggf. an das Skript übergeben (in REXX kann der Inhalt eines übergebenen Argumentes mit `loginarg= ARG(1)` abgefragt werden).

Die automatische Ausführung eines REXX-Programms kann z.B. dazu verwendet werden, um sich automatisch in einer Gegenstelle einzuloggen oder nach dem Login weitere Funktionen automatisiert auszuführen. Für einfache Login-Prozeduren (wie das Senden von Benutzer und Kennwort bei Telnetverbindungen) ist in der Regel jedoch die Verwendung der AutoLogin-Funktion (s.u.) unkomplizierter.

AutoLogin

In dieses Feld kann eine Ablaufsequenz (bestehend aus `Wait=` (warte auf) und `Send=` (sende) Sequenzen) eingetragen werden, um sich automatisch in eine Gegenstelle anzumelden. Da für die meisten Logins nur ein Ablauf in der Form 'Warte auf dies, dann sende jenes' benötigt wird, bietet ZOC diese Methode an, um ggf. auf die etwas komplexere REXX-Programm verzichten zu können.

Tasten wie Eingabe oder ESC können anhand Steuerzeichen angegeben werden (siehe [Steuerzeichen](#)). Die häufigsten Steuerzeichen sind `^[` für Esc, `^M` für Return, `^&` für das Kennwort und `^%` für den Benutzernamen (wenn diese oben im Dialog eingetragen sind).

Beispiel:

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

Dieses Beispiel bewirkt folgendes:

Warte bis die Gegenstelle den Text `Select Host:` schickt und simuliere dann die Tastatureingabe `Server` und die Eingabetaste Taste (durch den Code `^M`). Dann warte auf `login:` und sende als Antwort den Benutzern der oben im Dialog eingetragen ist und wieder die Eingabetaste. Danach warte auf den Text `password:` und sende das Kennwort und Return.

AutoLogin bei der nächsten Anmeldung aufzeichnen

AutoLogin Sequenzen müssen nicht von Hand geschrieben werden, sondern können durch die Funktion [AutoLogin bei der nächsten Anmeldung aufzeichnen](#) von ZOC automatisch erlernt werden. Hierzu setzen Sie den Haken im Dialog und tragen oben Ihren Benutzernamen und Kennwort ein. Dann speichern Sie den Eintrag und verbinden sich mit der Gegenstelle über das Hauptfenster des Adressbuchs und melden sich mit Benutzernamen und Kennwort an. Nachdem Sie angemeldet sind, wählen Sie im Skript Menü die Funktion [Aufzeichnung beenden](#). Damit wird das AutoLogin Feld von ZOC ausgefüllt und steht beim nächsten Verbindungsaufbau automatisch zur Verfügung.

1.5.1.4 FTP

Diese Einstellungsseite erlaubt es, eine FTP-Server-Verbindung einzurichten, die mit einer Gegenstelle des jeweiligen Adressbuch-Eintrages assoziiert ist. Das kann z.B. eine SFTP-Verbindung für eine Secure-Shell Verbindung sein, oder eine FTP-Verbindung, die zusammen mit einer zOS-Host-Verbindung aufgebaut wird.

Auf diese Weise wird der Austausch von Dateien deutlich vereinfacht, weil Dateien einfach per Drag-and-Drop übertragen werden können, statt mit SCP oder IND\$FILE Befehlen.

FTP-Typ

Dieses Feld ermöglicht Ihnen die Auswahl des vom Server angebotenen FTP-Verbindungstyps (z.B. SFTP (SSH) oder FTPS (SSL)) für das interne FTP-Fenster, das Teil des ZOC Terminals ist.

Alternativ können Sie einen externen FTP-Client (z.B. FileZilla oder WinSCP) verwenden. Der externe FTP-Client wird unter [Optionen → Programmeinstellungen → Spezielle Dateien](#) konfiguriert. Parameter zum Aufruf des externen FTP-Clients, um eine Verbindung zu diesem speziellen Server herzustellen, können unten angegeben werden.

Einen anderen Server für FTP-Verbindungen verwenden

Schalten Sie diese Option ein, wenn der FTP-Server, der mit dieser Verbindung assoziiert ist, eine andere Adresse besitzt, als für den Terminal-Zugriff verwendet wird.

Andere Zugangsdaten für die FTP-Authentifizierung verwenden

Wenn die Zugangsdaten des FTP-Servers sich von den Zugangsdaten des Verbindungseintrages unterscheiden, können hier ein alternativer Benutzer und Passwort angegeben werden.

Lokaler/Remote Start-Ordner

Diese Ordner-Angaben legen fest, in welchen Verzeichnissen sich das FTP-Fenster befindet, wenn es geöffnet wird. Der lokale Start-Ordner bezieht sich auf Ihren Computer (z.B.

`C:\users\you\Desktop`) und der Remote Start-Ordner auf die Gegenstelle (z.B. `/home/you/development`).

Parameter für ext. Server

Wenn Sie oben den FTP-Typ "Extern" wählen, können Sie hier einen Parameter eingeben, der an den externen FTP-Client weitergegeben wird, um anzugeben, wohin die Verbindung hergestellt werden soll. Typischerweise handelt es sich dabei um eine URL-ähnliche Zeichenfolge mit Benutzername, Passwort und Servername, z.B. `sftp://harry:secret@hogwarts.edu`.

Der Parameter kann eine oder mehrere der folgenden Platzhalter enthalten, die durch die jeweiligen Werte aus diesem Hostverzeichniseintrag ersetzt werden: `%USERNAME%`, `%PASSWORD%`, `%FTPSERVER%`.

Die [Standard](#)-Schaltfläche neben diesem Feld füllt es mit einem typischen SFTP-URL-Parameter für FileZilla oder WinSCP.

1.5.1.5 Verknüpfung

Diesen Adressbucheintrag auch im Datei-Menü des Hauptfensters anzeigen

Das Anwählen dieser Option erzeugt einen Eintrag im Datei-Menü von ZOC, so dass der Adressbucheintrag von dort aus schnell angewählt werden kann.

Verbindung per F-Taste ...

Hier kann eine Funktionstaste festgelegt werden, über die (zusammen mit der Strg-Taste, z.B. Strg+F6) eine Verbindung zu der Gegenstelle aufgebaut werden kann.

Verknüpfung auf der Arbeitsoberfläche erstellen ...

Dieser Knopf erstellt ein Icon mit dem Aufruf von ZOC und gleichzeitigem Verbindungsaufbau zu dieser Gegenstelle auf der Arbeitsoberfläche.

1.5.1.6 Fenster

Für jeden Eintrag kann eingestellt werden, ob er im gleichen ZOC Fenster oder ob ggf. ein neues Registerblatt angelegt oder ein eigenes Fenster geöffnet werden soll.

Falls Sie für die Verbindung ein neues Registerblatt anlegen lassen, sollte der Eintrag auf [Einmaliger](#)

[Verbindungsversuch](#) (im Abschnitt [Gegenstelle](#) dieses Dialogs) gesetzt werden, da andernfalls bei gleichzeitiger Selektion mehrerer Einträge mit [Verbindungsdialog mit Wiederholung](#) der Verbindungsaufbau über eine Warteschlange geschieht (siehe Funktion [Verbinden](#) im [Adressbuchfenster](#)).

Für neue Fenster kann angegeben werden, ob das Fenster immer an der gleichen Position am Bildschirm geöffnet werden soll (die Breite kann allerdings abhängig von den gewählten Zeichensatz und Fenstereinstellungen, variieren). Die Position wird in Bildpunkten am Bildschirm angegeben, wobei einfach die Werte des aktuellen Fensters übernommen werden können. Die tatsächliche Fensterposition kann jedoch abweichen, da vor allem die Fenstergröße auch von den Einstellungen des verwendeten Verbindungsprofils beeinflusst wird.

1.5.1.7 Mailbox

Verbindung ist fällig alle ...

Wenn ZOC nach Anwahl einer Mailbox ein Carrier-Detect Signal vom Modem erhält, wird das Datum des letzten Anrufs in diesem Eintrag aktualisiert. Zusätzlich geschieht dies auch mit allen weiteren Einträgen, deren Name in den ersten sechs Buchstaben mit diesem Eintrag übereinstimmt (siehe oben). Auf diese Weise können Mailboxen mit mehreren Telefonnummern verarbeitet werden.

Soll eine Mailbox regelmäßig angewählt werden (z.B. einmal in der Woche), kann in diesem Feld die Anzahl der Tage angegeben werden, ab denen ZOC mit einem kleinen gelben Pfeil im Adressbuch daran erinnert, dass es Zeit ist, diese Nummer wieder einmal anzuwählen.

Optionen für IEMSI Login

Einige Mailboxen bieten für den automatischen Login IEMSI an.

Für diese Methode muss der Name in den IEMSI Optionen und das Kennwort im Adressbucheintrag angegeben werden. Zusätzlich können noch verschiedene Einstellungen für die spätere Bedienung der Mailbox angegeben werden.

Während des Logins sendet die Gegenstelle dann eine IEMSI Anforderung (**IEMSI_IRQ, nicht zu verwechseln mit **EMSI_REQ, das für FIDO Mailsoftware verwendet wird), worauf ZOC den Benutzern, das Kennwort und alle sonstigen IEMSI Einstellungen zur Gegenstelle sendet. Auf diese Weise werden alle Abfragen zum Login übersprungen.

1.5.1.8 Info

Anmerkungen zu dieser Hostverbindung

Hier kann pro Eintrag ein beliebiger (bis zu 60 Zeichen langer) Text gespeichert werden (der Name des Sysops, Login-Zeiten usw.). Für Kennwörter sollte die AutoLogin Funktion oder das Kennwortfeld verwendet werden.

Ist dieses Feld gefüllt, wird der Feldinhalt im Hauptfenster des Adressbuches statt der Anzahl der Anrufe angezeigt. Dies kann verhindert werden, wenn der Memotext mit einem Leerzeichen beginnt.

Die anderen Felder werden von ZOC gefüllt und können nicht verändert werden.

1.5.2 Adressbuch-Einstellungen

Angezeigte Spalten

Mit dieser Einstellung können Spalten des Adressbuchs ausgeblendet werden, wenn diese nicht benötigt werden.

Zuletzt selektierten Eintrag merken

Diese Einstellung stellt ein, ob beim Öffnen des Adressbuchs die Einträge, die beim letzten Schließen ausgewählt waren, automatisch erneut ausgewählt sind.

Tooltips für Einträge anzeigen

Wenn Sie mit der Maus über einem Eintrag im Adressbuch stehen bleiben, erscheint nach einiger Zeit ein Fenster mit zusätzlichen Informationen zum Eintrag (Tooltip). Falls gewünscht, können die Tooltips mit dieser Option deaktiviert werden.

1.5.3 Verknüpfung erstellen

Diese Funktion erlaubt es Ihnen, ein Icon auf der Arbeitsoberfläche oder ein Feld in den Benutzerknöpfen des aktiven Verbindungsprofils zu erstellen. Hierdurch können Sie häufig genutzte Verbindung ohne Umweg über das Adressbuch aufbauen.

1.5.4 Die AutoVerbindungs-Funktion

Werden im Adressbuch ein oder mehrere Einträge ausgewählt oder die Funktion 'Neue Verbindung' im [Datei-Menü](#) aufgerufen, erscheint das AutoWahl-Fenster. Es versucht, Verbindungen zu den angegebenen Telefonnummern oder Internetadressen herzustellen. Ist belegt, wird der nächste Host angewählt (bei mehreren ausgewählten) oder die Verbindung zum gleichen erneut angewählt. Die maximale Anzahl der Anwahlversuche sowie die Wartezeit zwischen den Versuchen kann in [Optionen→Programmeinstellungen→AutoWahl](#) angegeben werden.

Während der AutoWahl kann der Ablauf anhand von vier Knöpfen beeinflusst werden:

Abbruch

Beendet den gesamten AutoVerbindungs-Vorgang

Wiederholung

Wartet AutoVerbindung darauf, den nächsten Eintrag zu wählen, kann die eingestellte Wartezeit übersprungen und der nächste Versuch sofort unternommen werden.

Nächster

Diese Funktion überspringt den aktuellen Verbindungsaufbau und fährt mit dem nächste Eintrag fort. Der übersprungene Eintrag wird später bearbeitet.

Löschen

Diese Funktion überspringt den Anruf der aktuellen Nummer und wählt die nächste Nummer. Der übersprungene Eintrag wird aus der Liste gelöscht und nicht mehr angewählt.

Als besonderes Feature kann das Fenster zwischen einer kleinen und einer großen Version umgeschaltet werden, indem der Minimieren/Maximieren-Knopf in der rechten oberen Ecke des Fensters angewählt wird.

1.5.5 Importieren fremder Telefonbücher

Import aus anderen Terminal-Programmen

ZOC kann einige Adressbuchformate anderer Terminalprogramme (wie Putty, Telix für DOS und Windows, Telemate für DOS und Windows usw.) lesen und in das ZOC-Format importieren. Um ein Adressbuch zu importieren, muss der Adressbuchtyp, die Zielsektion im ZOC-Adressbuch sowie die zu importierende Datei angegeben werden. Nach Drücken von OK werden die im fremden Adressbuch enthaltenen Einträge in ZOC importiert.

Text Import

Um ein Adressbuch in einem nicht direkt unterstützen Format (Datenbank oder Tabellenkalkulation) zu importieren, muss eine Textdatei erzeugt werden, die dann vom CSV/Text-Import eingelesen werden kann. Eine derartige Textdatei enthält einen Eintrag pro Zeile, aufgeteilt in Spalten, die mit einem speziellen Zeichen voneinander getrennt sind (z.B. Komma, Strichpunkt, senkrechter Strich, etc.). Datenbanken und Tabellenkalkulationen sind in den meisten Fällen in der Lage, Daten in eine Datei mit CSV-Format zu exportieren, die für den ZOC Text-Import geeignet ist. Diese Datei benötigt zwingend Spalten für das **Name/Label** Feld und das **Verbinden mit** Feld, optional Spalten für Benutzername, Passwort und Port, egal in welcher Reihenfolge.

Beispiel:

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

Sonderlösungen

Möchten Sie Daten importieren, die über die Import-Funktionalität von ZOC hinausgehen, können Sie alternativ eine Datei im ZOC-eigenen Format erstellen. Da das ZOC-Adressbuch (HostDirectory.zhd) als reine Textdatei gespeichert wird, ist es mit vertretbarem Aufwand und einer Scriptsprache wie Visual Basic, Perl oder REXX problemlos möglich, die entsprechenden Adressbucheinträge im ZOC-eigenen Format bereitzustellen.

Korrekturen nach erfolgreichem Import

Das ZOC-Adressbuch ermöglicht das Markieren mehrerer Einträge und das gemeinsame Editieren häufig verwendeter Einstellungen dieser Einträge. Fehlende oder fehlerhafte Einstellungen können so meist problemlos ergänzt werden.

1.6 Weitere Funktionen

- ☐ [Benutzerdefinierte Aktionen](#)
- ☐ [Das Transfer-Fenster](#)
- ☐ [IND\\$FILE Dateiübertragung](#)
- ☐ [SCP Dateiübertragung](#)
- ☐ [Zmodem Dateiübertragung](#)
- ☐ [Das Datenprotokollfenster](#)

1.6.1 Benutzerdefinierte Aktionen

Benutzerdefinierte Schaltflächen, Tastaturtasten, Auto-Makros, Leerlaufaktionen und andere Ereignisse können so konfiguriert werden, dass sie Funktionen wie das Senden von Text, das Initiieren einer Verbindung, das Öffnen eines Dokuments usw. ausführen. In diesem Dialog können Sie Aktionen auswählen und zusätzliche Details dazu angeben, wie diese auszuführen sind.

Viele dieser Auswahlmöglichkeiten sind selbsterklärend, aber hier sind weitere Details für diejenigen, die es nicht sind:

Text an Gegenstelle senden

Der zu sendende Text kann Steuerzeichen aus der Liste der [Steuerzeichen](#) enthalten. Beispielsweise kann der Text `^M` für Enter oder `^I` für Tab oder `^Z` für die Enter/Submit-Taste in einer TN3270-Emulation enthalten.

(Hinweis: Für Emulationstasten, die kein entsprechendes Steuerzeichen haben, können Sie stattdessen die Aktion 'Sende eine Mischung aus Text und Emulationstasten' verwenden).

Sende eine Mischung aus Text und Emulationstasten

Diese Funktion ist nützlich in Fällen, in denen es notwendig ist, Kombinationen aus Text und speziellen Emulationstasten zu senden, die nicht in der Liste der [Steuerzeichen](#) enthalten sind.

Der Parameter für diese Aktion ist Text, der gesendet werden soll, aber er kann auch die [Tastenbezeichnungen der Emulationen](#) enthalten, die in spitzen Klammern erscheinen können. Zum Beispiel entspricht in dieser Aktion `Hello world!<Enter>` dem `Hello world!^M` in der Aktion [Text an Gegenstelle senden](#), aber dieses Format ist flexibler, da Sie alle verfügbaren [Tastenbezeichnungen der Emulationen](#) verwenden können, wie z.B. `<Home>submit<F3>save<Enter>`.

Hinweis: Wenn Sie tatsächlich die Zeichen `<` und `>` senden müssen, können Sie `<lt>` und `<gt>` als Platzhalter verwenden, z.B. `diff -u file1.dat file2.dat <gt>files.diff<Enter>` sendet `diff -u file1.dat file2.dat >files.diff` gefolgt von der Enter-Taste.

Sende eine emulationsspezifische Sondertaste

Diese Aktion sendet eine spezifische Emulationstaste, wie die VT220 [Do](#)-Taste oder die TN3270

Insert-Taste. Sie ist hauptsächlich für einfache Fälle gedacht, in denen Sie lediglich eine Emulationsfunktion, wie die TN3270 **PA1**-Taste auf eine Taste oder Tastenkombination auf Ihrer lokalen Tastatur zuordnen möchten. Sie können die Namen der Tasten für jede Emulation finden, indem Sie auf die '&threedots;'-Schaltfläche neben dem Tasten-Eingabefeld klicken oder in den [Tastenbezeichnungen der Emulationen](#) dieses Hilfetextes nachsehen.

(Hinweis: Um mehrere Emulationstasten oder eine Kombination aus normalem Text und Emulationstasten zu senden, verwenden Sie stattdessen die Aktion [Sende eine Mischung aus Text und Emulationstasten](#)).

Führe eine Funktion aus dem ZOC Hauptmenü aus

Es ist auch möglich, eine Funktion aus dem Hauptmenü auszuführen (z.B. um sie auf eine Schaltfläche oder eine andere Tastenkombination zuzuordnen). Dies wird erreicht, indem der Name der Menüfunktion in das Aktionsfeld eingegeben oder die Funktion durch Klicken auf die '&threedots;'-Schaltfläche ausgewählt wird.

Aufbau einer Verbindung zu einem Eintrag aus dem Adressbuch

Um sich mit Gegenstellen zu verbinden, die im Hostverzeichnis gespeichert sind, geben Sie den Namen des Hostverzeichniseintrags an oder wählen ihn aus dem Hostverzeichnis über die '&threedots;'-Schaltfläche aus.

Verbindungsaufbau mit beliebigem Ziel/Kommunikationsmethode

Diese Aktion initiiert eine Verbindung, die nicht im Hostverzeichnis aufgeführt ist. Der Parameter hat das Format `<Verbindungstyp>!<Ziel>`, wobei der Verbindungstyp einer der im Sitzungsprofil angezeigten Typen ist, während das Ziel vom Verbindungstyp abhängt (z.B. für eine Secure Shell-Verbindung wird es eine IP-Adresse oder ein Hostname und Port sein, für eine Unix Pipe würde es den Namen einer Datei annehmen, etc.). Beispiele sind `Secure Shell!joe@nowhere.com`, `Secure Shell!harry:secret123@hogwarts.com:10022`, `Telnet!3270.mainframe.com:10023`, `Serial/Direct!COM5`, etc.

Starte ein REXX-Skript in ZOC

Um ein REXX-Skript von einer Taste oder Schaltfläche zu starten, geben Sie den Namen der Skriptdatei in das Aktionsfeld ein, z.B. `script\myscript.zrx`, und klicken Sie auf die Schaltfläche, die dem Assistenten mitteilt, dass der Text im Feld der Name des auszuführenden REXX-Skripts ist.

1.6.2 Das Transferfenster

Das Transfer-Fenster enthält folgende Informationen:

Dateiname

Der Name der gerade übertragenen Datei (bei Xmodem-Downloads nicht bekannt).

Dateigröße

Die Größe der gerade übertragenen Datei (bei Xmodem-Downloads nicht bekannt).

Übertragen

Die Anzahl der bereits gesendeten bzw. empfangenen Zeichen.

Zeit

Die bisherige Dauer der Übertragung sowie eine Schätzung der Gesamtdauer (ist nur verfügbar, wenn die Größe der empfangenen Datei bekannt ist).

Geschw.

Die Durchschnittsgeschwindigkeit der bisherigen Übertragung. Bei seriellen Übertragungen sollte

diese in etwa ein Zehntel der Baudrate betragen (z.B. 960 Zchn./Sek. bei 9600 bps).

Überspr.

Diese Funktion ist nur in Zmodem verfügbar und überspringt die Übertragung der aktuellen Datei.

Hinweis: Diese Option funktioniert nicht mit allen Zmodem Implementierungen.

Nach Übertragung auflegen

Diese Option bewirkt, dass ZOC nach erfolgreicher Übertragung automatisch auflegt.

Hinweis: Es erfolgt keine korrekte Abmeldung. Dies kann zu Problemen mit der Gegenstelle führen.

Datei nach Transfer löschen

Mit dieser Option kann die gerade übertragene Datei nach fehlerfreier Übertragung automatisch gelöscht werden.

1.6.3 IND\$FILE Dateiübertragung

Das IND\$FILE Protokoll dient dem Austausch von Daten zwischen dem lokalen Computer und dem Server/Host unter Nutzung einer bestehenden TN3270 Verbindung. Hierzu wird während der Verbindung aus dem Transfer Menü der Punkt Upload oder Download gewählt um Daten zum Server zu schicken oder vom Server zu empfangen.

Vor dem Start wird der lokale und entfernte Dateiname festgelegt, wobei der entfernte Name in einem Format vorliegen muss, das der Host versteht (sequentielle oder PO Datei), z. B. `SOURCE.ASM(TEST01)` oder `'TXUSER.OUTPUT.LOG'` (im zweiten Fall incl. der Hochkommata, um die Angabe eines voll qualifizierten Namens anzuzeigen.)

Die Optionen geben an, wie die Datei bei der Übertragung umgesetzt werden soll, z.B. die Umsetzung von EBCDIC nach ASCII, die Verwendung von Zeilenendezeichen und ggf. die Umcodierung von ASCII Zeichen entsprechend der momentan in der TN3270 Emulation eingestellten Codepage.

Entsprechend der Einstellungen des Host-Typs kann der Transfer erfolgen, während der Cursor auf einem TSO Prompt (`READY` oder `ISPF Menü 6`), auf einem ISPF Befehlsprompt (`Command ==>` oder `Option ==>`) oder auf Eingabebereitschaft eines VMS Systems steht.

1.6.4 SCP Dateiübertragung

Die ZOC SCP-Dateiübertragung ist eine interaktive Variante des SSH `scp` Befehls.

SCP erlaubt den Download einzelner Dateien, mehrerer Dateien (durch Verwendung von Platzhaltern), Verzeichnissen und Verzeichnisbäumen.

Wichtig: SCP Dateiübertragungen finden immer im Kontext der ursprünglichen SSH-Verbindung statt. Befehle wie `su ssh` am entfernten Rechner führen in der Regel entweder zum Abbruch oder zur Übertragung auf den ursprünglich verbundenen Rechner.

Wenn kein alternativer Pfad im Feld `Quelle` angegeben wurde, werden die zu empfangenden Daten aus dem `Home` Verzeichnis des entfernten Benutzers kopiert (nicht aus dem aktuellen Arbeitsverzeichnis der Shell). Allerdings wird dieses Feld von ZOC bereits mit dem aktuellen Arbeitsverzeichnis der Gegenstelle vorbelegt, welches ZOC durch Ausführen des Befehls `pwd` von der Gegenstelle abfragt, so dass Downloads wenn dieses Feld nicht geändert wird, aus diesem Ordner erfolgen.

Bei nicht aktivierter Option `Rekursiv` wird der Inhalt von `Quelle` als einzelne Datei oder Datei mit Platzhalter interpretiert.

Ist die Option `Rekursiv` hingegen aktiv, wird der Inhalt von `Quelle` als Verzeichnis interpretiert (z.B. `~/Projekt/`) und alle Dateien dieses Verzeichnisses inkl. aller Unterverzeichnisse werden

gedownloadet.

Die empfangenen Dateien werden im [Zielordner](#) gespeichert, der von ZOC mit dem Standard Downloadverzeichnis vorgelegt ist. Das Standardverzeichnis für Downloads kann unter [Optionen→Programmeinstellungen→Dateiablage](#) geändert werden.

Sollten Dateierweiterungen eingehender Dateien mit den für den alternativen Downloadordner eingetragenen Dateierweiterungen übereinstimmen (einstellbar unter [Optionen→Verbindungsprofil→Transfer](#)), werden diese Dateien stattdessen im alternativen Downloadordner abgelegt (einstellbar unter [Optionen→Programmeinstellungen→Dateiablage](#)).

1.6.5 Zmodem Dateiübertragung

Überblick

Die meisten Unix/Linux Server verfügen über Zmodem Programme (rz und sz bzw. lrz und lsz), die Dateitransfer von und zu entfernten Rechnern mit ZOC auf elegante Weise ermöglichen.

Falls Zmodem auf einem Server nicht installiert ist, suchen Sie im Internet nach lrzsz.zip, das dann auf die unter Unix übliche Art und Weise (./configure, make, make install) installiert werden kann. Alternativ bieten viele Linux Distributionen ein entsprechendes Paket an.

Transfer

Um eine Datei vom Host zu ZOC zu senden, geben Sie auf der Linux-Shell den Befehl `sz <dateiname>` ein. ZOC öffnet dann einen Zmodem Download und speichert die Datei im Download-Verzeichnis (siehe [Optionen→Programmeinstellungen→Dateiablage](#)).

Um eine Datei zum Unix/Linux Rechner zu schicken, starten Sie unter ZOC im Transfer Menü einen Zmodem Upload während Sie Linux-seitig in einem Befehlsprompt stehen. Daraufhin schickt ZOC zunächst den Befehl `rz` um unter Linux das Empfangsprogramm zu starten und führt dann auf Ihrem Rechner den Upload durch. Alternativ können Sie auf der Linux-Shell auch `rz` wodurch der Upload in ZOC automatisch gestartet wird.

Optionen

Für die Dateiübertragung per ZModem stehen im Verbindungsprofil (Abschnitt Transfer) die folgenden Einstellungen zur Verfügung:

Upload Datei-Management

Diese Optionen sind nur bei Uploads wirksam. Sie funktionieren nur dann, wenn auch der Empfänger diese Optionen verarbeitet bzw. verarbeiten will. Die DSZ- bzw. RZ-Implementationen von Zmodem auf verschiedensten Plattformen unterstützen diese Funktionen.

Es kann angegeben werden, ob der Empfänger CRs und LFs seinem Betriebssystem entsprechend anpassen soll, und was geschehen soll, wenn eine gesendete Datei beim Empfänger bereits vorhanden ist (die Unix SZ-Optionen sind -p, -y, -n, -r, -+).

ASCII

Die ASCII Option wird benutzt, um das Zeilenendezeichen zwischen verschiedenen Betriebssystemen umzusetzen. Der Einsatz ist nur bei Textdateien sinnvoll, Binärdateien (z.B. EXE oder ZIP) werden normalerweise bei Übertragung mit dieser Option unbrauchbar.

Datum/Zeit lokal

Wenn diese Einstellung aktiviert ist, erhält eine empfangene Datei das Datum und die Uhrzeit, zu der sie empfangen wurde und nicht die Werte, die die Gegenstelle hierfür sendet.

1.6.6 Das Datenprotokollfenster

Das Datenprotokoll-Fenster ermöglicht es Ihnen, durch den Text zu scrollen, den Sie seit dem Start von

ZOC (oder seit dem Löschen des Zurückblättern-Buffers) empfangen haben. Sie können es auch verwenden, um Debugging- und Datentrace-Ausgaben anzuzeigen, die unter [Trace/Debug](#) konfiguriert werden können.

Sie können Text mit der Maus markieren (und damit in die Zwischenablage kopieren) oder nach Text suchen, indem Sie die Taste F(inden) drücken.

1.7 ZOC konfigurieren

Es gibt verschiedene Arten von Optionen, die es erlauben, ZOC auf verschiedenen Ebenen zu konfigurieren: [Programmeinstellungen](#), [Allgemeine Programmeinstellungen](#) und [Verbindungsprofile](#). Außerdem gibt es noch die Konfigurationsmöglichkeit über die [commandline.ini Datei](#)

Die Programmeinstellungen bleiben im Gegensatz zu den Verbindungs-Einstellungen für alle Gegenstellen gleich. Aus diesem Grund gibt es auch nur einen Satz dieser Einstellungen.

Verbindungsprofile enthalten dagegen Gruppen von Einstellungen, die sich zwischen verschiedenen Gegenstellen unterscheiden können. Zum Beispiel ist es wahrscheinlich, dass für eine Verbindung zu einem IBM Großrechner eine andere Belegung der Knopfleiste oder Fenstergröße erwünscht ist, als bei einer Verbindung zu einem Linux Server. Das Verbindungsprofil gruppiert diese Einstellungen in einer Datei, die dann im Adressbuch der Verbindung zugeordnet werden kann.

Zusätzlich können für den Dialog Neue Verbindung und im Adressbuch noch separate Einstellungen vorgenommen werden, die die entsprechenden Einstellungen des Verbindungsprofils überlagern. Hierdurch kann das gleiche Verbindungsprofil (F-Tasten, Knöpfe, Bildelemente) für Verbindungen mit unterschiedlichen Emulationen und Verbindungstypen verwendet werden.

Siehe auch: [Optionen Überblick](#), [Optionen-Menü](#), [ZOC Konfigurieren](#), [Verbindungsprofile](#), [Tastaturprofile/Umsetzungsprofile](#), [Programmeinstellungen](#)

1.7.1 Programmeinstellungen

Programmeinstellungen sind im Gegensatz zu [Verbindungsprofilen](#) für alle Verbindungen gültig. Sie enthalten Einstellungen, die sich kaum noch verändern, nachdem sie einmal eingestellt sind.

Programmeinstellungen gibt es für die Bereiche

- ☐ [Fenster](#)
- ☐ [Registerblätter](#)
- ☐ [Maus](#)
- ☐ [Zwischenablage](#)
- ☐ [Zurückblättern](#)
- ☐ [Chat Modus](#)
- ☐ [Drucker](#)
- ☐ [Klänge](#)
- ☐ [Abfragen](#)
- ☐ [Gesp. Kennwörter](#)
- ☐ [Dateiablage](#)
- ☐ [Spez. Dateien](#)
- ☐ [Neue Verbindung](#)
- ☐ [AutoWahl](#)
- ☐ [Verbindungsende](#)
- ☐ [Sonstiges](#)

- ☐ [Features](#)
- ☐ [Updates](#)
- ☐ [ZOC Kennwort](#)
- ☐ [Sprache](#)

Siehe auch: [Optionen Überblick](#), [Optionen-Menü](#), [ZOC Konfigurieren](#), [Tastatur/Umsetzungsprofile](#), [Verbindungsprofile](#).

1.7.1.1 **Fenster**

Fenstertitel

Dieses Feld kann einen Text festlegen, der im Titelbalken des Hauptfensters angezeigt wird. Der Text kann Platzhalter enthalten, die von ZOC gegen die später jeweils passenden Werte ersetzt werden, z.B. `ZOC %VERSION% %CONNECTEDTOHOST% %OPTIONS%` oder auch `%COMPUTERNAME% %CONNECTEDTOHOST% %OPTIONS%`.

%CONNECTEDTOHOST%

Der Text 'verbunden mit' sowie der Name der Gegenstelle.

%COMPUTERNAME%

Der Name des Computers.

%HOST%

Der Name der Gegenstelle (z.B. Titel des Adressbucheintrags).

%HOSTRAW%

Der Inhalt des Verbinden-mit Feldes.

%INSTANCE%

Die lfd. Nummer des momentan offenen ZOC Fensters.

%OPTIONS%

Der Name des aktuell verwendeten Verbindungsprofils.

%VERSION%

Die Versionsnummer von ZOC.

%ZOCORHOST%

Wenn nicht online, enthält dieser Platzhalter den Text "ZOC", wenn online, den Namen der Gegenstelle.

%ZOCVERSORHOST%

Wie `%ZOCORHOST%` aber incl. der ZOC-Versionsnummer.

Maximierter Fensterrand in Farbe

Wenn das Programmfenster maximiert ist, gibt es einen Bereich, der die Lücke zwischen dem Terminalbereich und dem Rand des Fensters füllt (der Grund dafür können zum Beispiel Einstellungen im Terminal-Layout sein, wie unter [Verbindungsprofil→Layout](#) beschrieben). Normalerweise ist dieser Bereich hellgrau eingefärbt, kann aber durch diese Option auch in einer anderen Farbe dargestellt werden.

Zusätzliche Terminal-Umrandung

Diese Option zeichnet eine Umrandung um den Terminalbereich, um diesen optisch vom Fensterrad und dem Text abzuheben. Als Farbe für diese Umrandung wird die Hintergrundfarbe des Verbindungsprofils verwendet.

Fenstertransparenz

Wenn diese Option auf Werte größer Null eingestellt ist, wird das Programmfenster teilweise transparent. Bitte beachten Sie jedoch, dass dies mit älteren Grafikkarten zu **erheblicher zusätzlicher Systemlast** führen kann.

Siehe auch: [Erweiterte Fenster-Einstellungen](#), [Programmeinstellungen](#)

1.7.1.2 Erweiterte Fenster-Einstellungen

Programmfenster während des Verbindungsaufbaus verkleinern

Diese Einstellung hat zur Folge, dass während der Anwahl von Gegenstellen das Hauptfenster von ZOC minimiert wird und nur das AutoWahl-Fenster sichtbar bleibt. Nach erfolgreicher Anwahl oder dem letzten erfolglosen Versuch wird die alte Fenstergröße automatisch wiederhergestellt.

Fenster nach Verbindungsaufbau in den Vordergrund bringen

Diese Option bewirkt, dass das ZOC-Fenster bei beim Zustandekommen einer Verbindung zur Gegenstelle nach vorne kommt. Auf diese Weise kann ZOC im Hintergrund anwählen und kommt erst nach erfolgreichem Verbindungsaufbau nach vorne.

Fenster während des Datei-Transfers verkleinern

Bei aktivierter Funktion minimiert sich das ZOC-Fenster während einer Dateiübertragung, so dass nur das Transferfenster sichtbar bleibt.

Fenster nach Datei-Transfer in den Vordergrund bringen

Wenn diese Option aktiviert ist, kommt das ZOC-Fenster nach Beenden einer Dateiübertragung automatisch nach vorne. Auf diese Weise kann während einer Dateiübertragung gearbeitet werden, ohne dass das Ende der Übertragung verpasst wird.

Standard- oder Ersatz-Schriftart

Mit dieser Option kann eine Standard-Schriftart für ZOC angegeben werden. Diese wird immer dann verwendet, wenn ZOC nicht in der Lage ist, eine speziell angegebene Schriftart zu öffnen. Diese Situation kann z.B. auftreten, wenn ZOC-Konfigurationsdateien (z.B. Verbindungsprofile) zwischen Usern, die auf unterschiedlichen Systemen arbeiten, ausgetauscht werden, bzw. falls solche Dateien zwischen der Windows und macOS Version von ZOC transferiert werden.

Siehe auch: [Fenster](#), [Programmeinstellungen](#)

1.7.1.3 Registerblätter

Aktion beim Öffnen neuer Registerblätter

Die hier eingestellte Aktion wird ausgeführt, wenn ein *zusätzliches* Registerblatt durch eine Benutzeraktion geöffnet wird (z.B. über das Menü). Zur Wahl stehen die Anzeige des Dialogs [Neue Verbindung](#) oder des [Adressbuches](#), sowie die Ausführung des Standard-Scripts `Startup.zrx` welchem dann zusätzlich ein Parameter `##NEWTAB##` mit übergeben wird, um anzuzeigen, dass dieses Script durch ein neues Registerblatt angestoßen wurde.

Schließen-Schaltfläche in Registern

Mit dieser Option können Sie festlegen, für welche der Registerblätter ein Schließen-Knopf angezeigt wird.

Registerblätter immer anzeigen

Falls mehr als eine Verbindung in einem ZOC-Fenster geöffnet ist, zeigt das Programm eine Leiste mit Registerblättern für jede Verbindung an. Diese Option legt fest, ob diese Leiste auch dann noch angezeigt werden soll, wenn nur eine einzige Verbindung geöffnet ist.

Registerblätter mit Alt+1 - Alt+9 aktivieren

Diese Einstellung ermöglicht es, per Tastenkombination zwischen den ersten neun Registerblättern wechseln zu können. Die Tastenkombination unter Windows ist hierfür Alt+Ziffer, unter macOS ist es Cmd+Ziffer.

Tabs durch Xterm Sequenz für Fenstertitel umbenennen

Gegenstellen können eine Sequenz senden, um Text in der Titelzeile des Terminal-Fensters anzuzeigen. Wenn diese Option aktiviert ist, benennt diese Sequenz zusätzlich auch noch den Namen des entsprechenden Registerblattes um.

Tabs mit Mausrad wechseln, wenn die Maus in der Tabs-Leiste steht

Wenn der Mauszeiger im Bereich der Registerblätter steht, kann man mit dem Mausrad zum nächsten bzw. vorherigen Tab wechseln.

Farbige Registerblätter

Mit dieser Funktion wird die Farbe aus der Farbmarkierung des Registerblatts zum Einfärben des Registerblattes und als Hintergrund für Benutzerknöpfe, Chatzeile und Statuszeile verwendet. Damit wird der Wechsel zwischen Verbindungen mit unterschiedlicher Farbmarkierung am Bildschirm deutlicher.

Die Farbe für die Registerblätter usw. wird ggf. aus dem Adressbuch übernommen, wenn dort für den gewählten Eintrag ein Farbsymbol zugeordnet ist.

Wenn diese Option aktiviert ist und wenn jedoch keine Tabs angezeigt werden (wenn nur eine Verbindung im Fenster ist) werden die Fensterelemente nur dann eingefärbt, wenn der Verbindung eine spezifische die Farbe aus dem Adressbuch zugeordnet wurde.

Siehe auch: [Programmeinstellungen](#)

1.7.1.4 Maus

Funktion der Mausknöpfe

Für die linke, mittlere und rechte Maustaste kann getrennt festgelegt werden, welche Funktion mit Klick der Taste ausgeführt werden soll.

Die Auswahlpunkte, die sich auf Markierung/Kopieren beziehen, werden von der Option zum [Text nach dem Markieren automatisch kopieren](#) im Abschnitt [Zwischenablage](#) beeinflusst. Die Auswahl [Sende Enter](#), wenn sie in einer TN3270- und TN5250-Sitzung aufgerufen wird, bewegt den Cursor vor dem Senden von Enter an die angeklickte Stelle und ermöglicht so zum Beispiel die Auswahl eines ISPF-Menüs mit der Maus.

Funktion des Mausrads

Diese Einstellung bietet eine Auswahlliste zur Funktionalität des Mausrads. Die meisten Optionen sind offensichtlich, mit Ausnahme von der Option Smart. In dieser Einstellung scrollt das Mausrad in den Zurückblättern-Bereich, außer eine Remote-Anwendung (z.B. Linux VIM, screen, MC, usw.) hat das Terminal in den Alternate-Screen Modus geschaltet. In diesem Fall bewegt das Mausrad den Cursor. Im Normalfall scrollt das Mausrad also in den Zurückblättern-Bereich, solange Sie sich in einer Linux-Shell befinden, bewegt aber den Cursor, wenn Sie sich in einer Bildschirm-basierten Anwendung befinden.

Begrenzungszeichen

Wenn mit einem Doppelklick auf den Bildschirm ein Wort markiert wird, entscheidet die Liste der Begrenzungszeichen, welche Zeichen noch als Teil des Wortes angesehen werden, und welche die Markierung begrenzen.

Siehe auch: [Programmeinstellungen](#)

1.7.1.5 Zwischenablage

Text nach dem Markieren automatisch kopieren

Wenn diese Option aktiviert ist, wird markierter Text sofort in die Zwischenablage kopiert.

Text-Markierung nach dem Kopieren entfernen

Bei aktivierter Option wird die Markierung eines Textes aufgehoben, nachdem der markierte Text in die Zwischenablage kopiert wurde.

Unterdrücken der Meldung 'Text wurde kopiert'

Mit dieser Option kann die Bestätigungsmeldung unterdrückt werden, die ZOC normalerweise beim Kopieren in die Zwischenablage anzeigt.

Strg+C / Strg+V für Kopieren und Einfügen verwenden

Die Tastenkombinationen Strg+C und Strg+V werden in Terminalprogrammen normalerweise dazu verwendet, um einen bestimmten Code (Hex 16 und Hex 03) an die Gegenstelle zu senden.

Falls Sie diese Funktion nicht benötigen, können Sie mit Hilfe dieser Option beim Drücken von Strg+V stattdessen den Text aus der Zwischenablage senden und bei markiertem Text am Bildschirm mit Strg+C den Text in die Zwischenablage kopieren. **Hinweis:** Wenn kein Text markiert ist funktioniert Strg+C wie gewohnt zum Senden des entsprechenden Steuerzeichens.

Strg+Shift+C / Strg+Shift+V für Kopieren und Einfügen verwenden

Ähnlich der vorherigen Funktion, aber es muss zusätzlich noch die Umschalt-Taste gedrückt werden. Strg+C und Strg+V werden somit als Terminal-Codes gesendet, aber zusammen mit gedrückter Umschalt-Taste wird auf die Zwischenablage zugegriffen.

Funktion der Alt-Taste beim Markieren umkehren

Normalerweise wird Text im Fließmodus markiert. Gleichzeitiges Drücken der Alt-Taste markiert Text stattdessen im Blockmodus. Bei aktivierter Option wird dieses Verhalten vertauscht.

Einzeilige Markierung sofort senden

Diese Funktion bewirkt, dass ein am Bildschirm markierter Text sofort gesendet wird, wenn er sich nur über eine Bildschirmzeile erstreckt. Alternativ kann ein einzeiliger markierter Text sofort gesendet werden, wenn diese Option halb gesetzt ist (graues Feld statt Haken) und wenn während des Markierens die Shift-Taste gedrückt gehalten wird. Wenn die Option komplett deaktiviert ist, wird einzeiliger Text nie automatisch gesendet.

Leerzeichen am Ende der letzten Zeile einer Markierung löschen

Diese Option löscht Leerzeichen am Zeilenende der letzten Zeile einer Textmarkierung.

Leerzeichen an allen Zeilenenden einer Blockmarkierung löschen

Diese Funktion löscht Leerzeichen, die bei einer Blockmarkierung am Ende einer jeden Zeile auftreten können.

Siehe auch: [Programmeinstellungen](#), [Bearbeiten-Menü](#)

1.7.1.6 Zurückblättern

Zurückblättern

Hier werden die Einstellungen für das Zurückblättern im ZOC Fenster (bzw. in einem eigenen Fenster) festgelegt. Die Festlegung der Größe geschieht in Zeilen, wobei eine große Anzahl von Zeilen viel Hauptspeicher belegen und die Performance beim Scrollen geringfügig beeinträchtigen wird.

Unter Linux gibt es Programme die in einen alternativen Bildschirmpuffer schreiben (z.B. VI, Midnight Commander, etc.). Diese generieren in der Regel Ausgaben die dazu führen, dass die Daten beim Zurückblättern unübersichtlich werden. Eine Ausnahme hierzu ist das Programm 'screen', das zwar auch in den alternativen Puffer schreibt, dessen Ausgabe im Zurückblättern aber wünschenswert sein kann. Deshalb gibt es eine Option, die den alternativen Bildschirminhalt beim Zurückblättern mit einschließt.

Einige Programme löschen den Bildschirm, indem nacheinander einzelne Zeilen gelöscht werden. Falls Ihnen Daten im Rückblättern-Puffer fehlen, könnte es helfen die Option zu aktivieren, die einzelne gelöschte Zeilen übernimmt. Falls Sie andererseits beim Zurückblättern einzelne Zeilen finden, die dort außer der Reihe auftauchen, sollten Sie versuchen Sie, diese Option zu deaktivieren.

Das Zurückblättern kann neben dem Hauptfenster auch in einem eigenen Fenster passieren (Alt+BildAuf) was den Vorteil hat, dass Sie den aktuellen Bildschirm und alte Daten gleichzeitig sehen können. Für letzteres können Sie auch eine eigene Schriftart festlegen.

Datenprotokollfenster

Der Datenprotokollpuffer speichert die ankommenden Zeichen in genau der Reihenfolge wie sie empfangen wurden. Hierbei können Zeilen die am rechten Rand umgebrochen wurden wieder gestreckt werden, was z.B. bei der Anzeige von Unix-Dateien hilfreich sein kann. Die Größe des dafür verwendeten Speichers wird in Kilobyte eingestellt.

Außerdem verwendet die Anzeige des Datenprotokollpuffers die gleiche Schriftart, die oben für das Zurückblättern eingestellt wurde.

Siehe auch: [Programmeinstellungen](#), [Anzeige-Menü](#)

1.7.1.7 Chat Modus

Schriftart auswählen

Hier kann der im Chat Modus Eingabefeld verwendete Zeichensatz und dessen Größe eingestellt werden.

Verwendung von Steuerzeichen als Eingabe zulassen

Bei aktivierter Option werden Steuerzeichen wie z.B. ^M aufgelöst (es wird ein Return gesendet). Andernfalls wird der Text als Text (es wird der Text ^M gesendet) übertragen.

Siehe auch: [Programmeinstellungen](#)

1.7.1.8 **Drucker**

Drucker

Hier kann der von ZOC verwendete Drucker bzw. Druckerport (z.B. LPT1), bzw. unter Windows Druck auf den Standarddrucker eingestellt werden.

Falls sie eine Hostanwendung nutzen, die direkt druckt, hängt die Art des zu verwendenden Druckers von der Art der Anwendung ab. Falls die Anwendung nur Text schickt, können Sie ZOC entweder auf einen Druckerport (LPTx) oder den für Sie passenden Drucker einstellen.

Falls die Anwendung jedoch auch Drucksteuerzeichen schickt, die genau auf Ihren Drucker abgestimmt sind (transparenter Druck), müssen Sie entweder einen Druckerport (LPTx) einstellen, bzw. den passenden Drucker auswählen und dann die Option [Druckertreiber umgehen](#) (s.u.) aktivieren.

Druckertreiber umgehen

Mit dieser Option ist es möglich den Druckertreiber zu umgehen und Druckdaten direkt zum Drucker zu schicken. Dies kann notwendig sein wenn der Server Daten mit Druckersteuerzeichen zum Drucker schicken will, z.B. um Seiten zu formatieren und Formulare zu drucken.

Einstellungen für den Direktdruck

Falls notwendig, kann ZOC so konfiguriert werden, dass bei Umgehung des Druckertreibers (s.o), vor und nach den eigentlichen Druckdaten noch eine einleitende und abschließende Datensequenz geschickt wird.

Diese Sequenzen können ZOC-Steuerzeichen enthalten (z.B. `^(1B)` oder `^[` für ein ESC-Zeichen) um druckerspezifische Codes an den Drucker zu schicken, z.B. um den Drucker zu initialisieren oder um auf einen anderen Papierschacht zu wechseln.

In-Datei Druckeinstellungen

Wird als Zieldrucker "In Datei" ausgewählt, kann beim Druck der Name der Zieldatei angegeben werden. Diese Datei erhält dann die ausgegebenen Druckdaten. Der Dateiname kann [Platzhaltercodes](#) enthalten, um Uhrzeit, Datum oder den Sitzungsnamen in den Dateinamen einzufügen.

Im Eingabefeld Post-Processing kann ein Befehl angegeben werden, der nach Abschluss des Druckauftrages ausgeführt wird. Der Platzhalter `%SPOOLFILE%` oder `$(SPOOLFILE)` kann verwendet werden, um den Dateinamen der Druckdatei im Befehlsaufruf anzugeben, z.B. `"C:\WINDOWS\system32\notepad.exe" "%SPOOLFILE%"` oder `/usr/bin/open -t "$(SPOOLFILE)"`.

Job-Verkettung

Wenn per VT-102 Sequenzen auf einen lokalen Drucker gedruckt wird, senden manche Programme die Start/Stop-Kennung nach jeder Zeile. Windows druckt dann jede Zeile auf eine neue Seite.

Die Option Druckjobverkettung steuert die Wartezeit (in Sekunden), die ZOC maximal verwendet, um einzelne Druckjobs noch mit dem vorherigen Job zu verketteten (0 schaltet die Verkettung ab).

Seitenkopf

Im Feld Seitenkopf können Sie eine Kopfzeile für die gedruckten Seiten definieren. Die Zeile kann aus drei Teilen bestehen, die durch senkrechte Striche getrennt und links, mittig und rechts ausgerichtet werden, z.B. `<links>|<mitte>|<rechts>`

In jedem dieser Teile können die Makros `%USERNAME%`, `%COMPUTERNAME%`, `%DATE%`, `%TIME%`, `%PAGE%`, `%CONNECTEDTO%`, `%TABNAME%` vorkommen, die dann durch die jeweiligen Daten ersetzt werden.

Beispiel: %COMPUTERNAME%|%USERNAME%|Seite %PAGE% würde den Rechnernamen links, den Namen des gerade angemeldeten Benutzers in die Mitte und einen Seitenzähler nach rechts in die Kopfzeile drucken.

Siehe auch: [Programmeinstellungen](#)

1.7.1.9 Klänge

Benachrichtigung

Hier können WAV-Dateien angegeben werden, die bei verschiedenen Ereignissen (z.B. Verbindungsende) abgespielt werden.

Soll kein Klang abgespielt werden, können die Felder leer gelassen oder eine nicht existierende Datei angegeben werden.

Piepton der Gegenstelle

Viele Gegenstellen verwenden ein akustisches Signal, um auf Fehleingaben oder besondere Situationen aufmerksam zu machen. Durch diese Option kann dieses Signal entweder in einen Piepton des internen Lautsprechers, in eine Klangdatei oder in bestimmte visuelle Effekte umgesetzt werden.

Zudem kann eine Obergrenze für die Anzahl der Pieptöne innerhalb einer bestimmten Zeit festgelegt werden. Wird die Anzahl überschritten unterdrückt ZOC die Signale. Dann muss wieder die gleiche Zeit ohne Signal verstreichen, bis ZOC die Pieptöne wieder aktiviert.

Siehe auch: [Programmeinstellungen](#)

1.7.1.10 Abfragen

Hier können Sicherheitsabfragen aktiviert werden, die vor 'gefährlichen' Operationen warnen.

Sicherheitsabfrage bei ...

Hier können Sicherheitsabfragen ('Sind Sie sicher?') für die angegebenen Aktionen aktiviert werden.

Warnung bei fehlendem Modem (kein CTS Signal)

Die meisten seriell angeschlossenen Geräte können anhand des CTS Signals am seriellen Port erkannt werden. Falls Sie Geräte haben, die kein CTS Signal liefern, kann hier die entsprechende Warnung deaktiviert werden.

Note: Diese Einstellung betrifft nur die Kommunikationsmethoden [Serial/Modem](#) und [Serial/Direct](#).

Warnung bei schneller Datei-Übertragung per Modem ohne RTS/CTS

Die RTS/CTS Steuerung ermöglicht eine sichere Kontrolle des Datenflusses zwischen Computer und angeschlossenem Gerät. Bei seriellen Übertragungen mit hoher Baudrate ohne RTS/CTS Steuerung können sehr leicht Übertragungsfehler auftreten. Bei aktivierter Funktion öffnet ZOC eine Meldung, falls eine Übertragungsrate größer 4800 Baud angewählt wird, ohne dass gleichzeitig RTS/CTS aktiv ist.

Note: Diese Einstellung betrifft nur die Kommunikationsmethoden [Serial/Modem](#) und [Serial/Direct](#).

Schutz vor Unterbrechung von Aktivität

Mit diesen beiden Einstellungen kann gewährleistet werden, dass der Benutzer nicht versehentlich

das Programm beendet, während noch Aktivitäten laufen.

Siehe auch: [Programmeinstellungen](#)

1.7.1.11 *Gespeicherte Kennwörter*

Kennwortspeicher

Die Optionen hier beziehen sich auf Kennwörter, die in den verschiedenen Kennwort-Abfragen durch Aktivieren von [Kennwort speichern](#) abgelegt wurden.

Kennwörter automatisch löschen

Sie können hier eine max. Lebenszeit der Kennwörter bei Nichtbenutzung angeben und ggf. eins oder mehrere dieser Kennwörter löschen.

Siehe auch: [Programmeinstellungen](#)

1.7.1.12 *Dateiablage*

REXX Skript

In diesem Ordner werden standardmäßig alle REXX-Dateien gesucht.

Profile

Dieses Verzeichnis dient der Speicherung von ZOC Verbindungsprofilen und anderen Einstellungsdateien (Tastaturbelegungen, Umsetzungstabellen, Programmeinstellungen, usw.).

SSH Dateien

In diesem Datenordner werden Dateien abgelegt, die mit SSH Verbindungen zu tun haben. Hierzu gehören Schlüsseldateien (Public/Private Keys), Tunnelprofile, die Liste bekannter Gegenstellen (known_hosts).

Download

Hier wird der Ordner angegeben, in dem normalerweise empfangene Download-Dateien gespeichert werden.

Alt. Download

Hier wird das Verzeichnis angegeben, in das spezielle Download-Dateien gespeichert werden. Die Dateiendungen der Dateien, die in dieses Verzeichnis umgeleitet werden sollen, werden im Dialog [Optionen→Verbindungsprofil→Dateibehandlung](#) festgelegt. Dieses Verzeichnis wird normalerweise dazu verwendet, um bestimmte Dateien beim Download in ein anderes Verzeichnis umzuleiten (z.B. Bilder).

Upload

Aus dem hier angegebenen Datenordner werden Dateien zum Upload gelesen, wenn nicht anderweitig ein anderer Ordner als Quelle angegeben wurde.

Text Upload

Aus diesem Verzeichnis werden die Dateien für die Funktionen [Transfer→Senden Text-Datei](#) und [Transfer→Senden Binär-Datei](#) gelesen.

Logdateien

Die Logdateien aus dem Log-Menü werden hier gespeichert.

Siehe auch: [Programmeinstellungen](#)

1.7.1.13 **Spezielle Dateien**

Editor

Dieses Feld enthält den Namen des Texteditors, den ZOC zum Bearbeiten von Dateien verwenden soll (z.B. beim Editieren eines REXX Skripts).

Hinweis: Wenn dieses Feld auf `<DEFAULT>` gesetzt ist, wird unter Windows der Editor [Notepad.exe](#) verwendet bzw. unter macOS der Standardeditor.

FTP-Client

Dieses Feld gibt einen externen FTP-Client an, der verwendet wird, wenn ein Host-Verzeichniseintrag so konfiguriert ist, dass er den externen FTP-Client anstelle des integrierten FTP-Fensters verwendet. Es sollte sich um einen vollqualifizierten Pfad zu einer ausführbaren Datei handeln, z.B. `C:\Program Files\FileZilla FTP Client\filezilla.exe` oder `/Applications/FileZilla/Contents/macOS/filezilla`. Diese ausführbare Datei wird dann mit einem host-spezifischen Parameter aufgerufen, der im FTP-Tab eines Eintrags im Adressbuch definiert werden kann.

Shell

Shell enthält den Namen der Shell, die bei Aufruf der Funktionen [Shell](#) im [Anzeigen-Menü](#) geöffnet wird. Diese Shell auch wird (zusammen mit der /C Option) verwendet, wenn innerhalb eines Skripts ein [ZocShell](#) Kommando ausgeführt oder eine Taste mit einer Shell Aktion belegt wird.

Hinweis: Wenn dieses Feld auf `<DEFAULT>` gesetzt ist, wird `cmd.exe` für Windows verwendet, sowie `/bin/sh` für macOS.

Verbindungsprotokoll

In die hier angegebene Datei protokolliert ZOC eine Übersicht aller Verbindungen und Dateiübertragungen (C+/C- bedeutet Connect/Disconnect, DL bedeutet Download, UL bedeutet Upload).

Der Dateiname kann Platzhalter wie `calls^2^3.LOG` für Datumsteile enthalten (siehe [Platzhalter](#) im Anhang).

Wenn kein Pfad angegeben wird, wird die Anrufprotokolldatei in den Log-Ordner von ZOC geschrieben. Wenn das Feld leer ist, wird kein Protokoll geschrieben.

Bildschirm speichern

Hier wird die Datei angegeben, in die bei Anruf der Funktion [Datei→Bildschirm speichern](#) der aktuelle Bildschirminhalt geschrieben wird.

ZOC-Ereignisse

Die hier angegebene REXX-Datei wird aufgerufen, wenn ZOC gestartet, wenn ZOC beendet und wenn innerhalb von ZOC die Kommunikationsmethode gewechselt wird. Technische Details dazu sind in der Beispieldatei `ZOCEVENT.ZRX` und in der Datei `ZOCEVENT.TXT` im Skriptverzeichnis zu finden.

Datei-Übertragung

Hier kann angegeben werden, welche REXX-Datei vor und nach einer Datei-Übertragung ausgeführt wird. Die REXX-Datei kann z.B. den Dateinamen der aktuellen Datei ändern, noch bevor diese auf Festplatte geschrieben wird (damit kann erreicht werden, dass Dateien je nach Endung in ein anderes Verzeichnis gespeichert werden). Außerdem können Packer und Entpacker (z.B. ZIP und UNZIP, UUdecode) noch vor Übertragung der Dateien automatisch aufgerufen werden.

Technische Details hierzu sind in der Beispieldatei `ZOCXFER.ZRX` und in der Datei `ZOCXFER.TXT` im Skriptverzeichnis zu finden.

Alternative REXX DLL

Standardmäßig liefert ZOC eine Ableger des Regina-REXX Prozessors mit. Falls Sie stattdessen eine andere REXX-Variante bevorzugen oder schon eine im Einsatz haben, können Sie hier deren DLL mit dem Programmier-API angeben. Wenn, wie z.B. bei OOREXX, das API auf mehrere DLLs verteilt ist, können Sie diese durch senkrechten Strich getrennt angeben (s.u.).

Das folgende ist eine Liste populärer REXX Implementierungen und der entsprechenden DLLs.

ZOC-REXX (akt. ZOC Standard)

`zocrexx.dll` (ZOC-REXX ist ein Ableger von REGINA REXX, siehe [Urheberrechte](#)).

Enterprise REXX (ZOC5 Standard)

`RxREXX.dll` (im Programmordner von ZOC5)

Regina-REXX

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

Object-REXX/OOREXX

`rexapi.dll|rexx.dll` (<http://www.oorexx.org>)

Reginald-REXX

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

Quercus Personal REXX

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

Wichtig: Bitte beachten Sie, dass es unter Umständen nötig sein kann, dass Sie statt nur des Namens, den gesamten Pfad zur entsprechenden DLL angeben, z.B.

`c:\Programme\oorexx\rexapi.dll|c:\Programme\oorexx\rexx.dll`

Siehe auch: [Programmeinstellungen](#)

1.7.1.14 Neue Verbindung

Die folgenden Einstellungen beziehen sich auf das Fenster [Neue Verbindung](#).

Bei manueller Verbindung den letzten Eintrag nicht vorauswählen

Mit dieser Einstellung lässt sich einstellen, ob das Fenster für neue Verbindungen die Adresse der letzten gemachten Verbindung wieder einblendet oder ob sich das Fenster mit einem leeren Eingabefeld öffnet.

Anzahl Einträge in Historie

Diese Einstellung legt die Anzahl der vorherigen Verbindungen fest, die durch Herunterklappen des Eingabefeldes wieder aufgerufen werden können.

Historie löschen

Durch diesen Knopf wird die Historie der letzten Verbindungen gelöscht.

1.7.1.15 *AutoWahl*

Die AutoWahl-Funktion wird beim Wählen aus dem [Adressbuch](#) verwendet, wenn die dortigen Einträge für AutoWahl konfiguriert sind.

Hinweis: Diese Funktion ist primär für Gegenstellen gedacht, die per Modem oder ISDN angewählt werden.

Die Funktion versucht, eine Gegenstelle anzuwählen und wiederholt diesen Anwahlversuch mehrmals. Die maximale Anzahl der Wählversuche sowie die Wartezeit zwischen den Anwahlversuchen der gleichen Telefonnummer kann eingestellt werden. Wenn diese Zeit auf z.B. 10 Sekunden eingestellt wird, drei Einträge im Adressbuch ausgewählt sind und alle drei Anwahlversuche scheitern, wird die erste Nummer sofort nach der dritten Nummer gewählt, da der Anwahlversuch für die zweite und dritte Nummer sicher länger als 10 Sekunden gedauert hat. Da einige Modems nach einem erfolglosen Anwahlversuch eine kurze Wartezeit fordern, kann zusätzlich noch die Mindestwartezeit zwischen jedem. Anwahlversuch angegeben werden.

Zusätzlich kann eine Zeichenkette definiert werden, die als Präfix für Anrufe per Serial/Modem oder ISDN verwendet wird. Dies ist nützlich, wenn z.B. eine 0 oder 9 vorgewählt werden muss, um ein Freizeichen zu erhalten. Wenn der Präfix nur für Serial/Modem benötigt wird, können stattdessen die Wählpräfixe unter [Modem Optionen](#) oder die Wähl-Makros (siehe unten) verwendet werden.

Kleines AutoWahl Fenster

Es kann zwischen einem großen und einem kleinen Fenster gewählt werden, das aktuelle Wählinformationen anzeigt. Das große Fenster verdeckt (naturgemäß) mehr Bildschirm, während das kleine Fenster lange Meldungen und Telefonnummer nicht vollständig anzeigt.

Hinweis: Zwischen großem und kleinem Fenster kann anhand des Knopfes Minimieren/Maximieren in der rechten oberen Ecke des Fensters umgeschaltet werden.

Wähl-Makros

Das Adressbuch bietet vier spezielle Makros für die Verwendung zusammen mit Telefonnummern an. Bei Verwendung der Zeichenketten \$1, \$2, \$3 oder \$4 irgendwo innerhalb der Telefonnummer (o.ä. für andere Verbindungsmethoden) ersetzt ZOC dieses Makro durch den unter Wählmakros gespeicherten Text.

Wenn Sie z.B. zur Anwahl einiger Adressbucheinträge Kreditkartennummern verwenden, können Sie \$1 mit 1234 5678 9012 3456 belegen und \$1 in jeder Telefonnummer angeben, in der diese Kreditkartennummer erscheinen soll.

Hinweis: Diese Makros funktionieren nur für Telefonnummern und können weder in Modem-Strings noch in Funktionstasten o.ä. verwendet werden.

Hinweis: In den meisten Fällen können ähnliche Ergebnisse durch Verwendung verschiedener Wählstrings (siehe [Modem Optionen](#)) erreicht werden.

Siehe auch: [AutoWahl Rückmeldungen](#), [Programmeinstellungen](#)

1.7.1.16 *AutoWahl Rückmeldungen*

Hier kann ZOC mitgeteilt werden, welche Meldungen das Modem und die anderen Kommunikationssysteme (Telnet, SSH, etc.) zurückliefern, um das Zustandekommen einer Verbindung, das Auftreten eines Fehlers, etc. zu melden.

Anhand der hier angegebenen Meldungen erkennen die Verbindungsfunktion von ZOC, ob tatsächlich eine Verbindung zustande gekommen ist, ob es Sinn macht, den Verbindungsversuch ggf. zu wiederholen oder den Verbindungsversuch als gescheitert zu betrachten.

Die Meldungen müssen mit einem senkrechten Strich (|) ohne Leerzeichen voneinander getrennt werden,

z.B. [CARRIER|CONNECT](#).

Siehe auch: [AutoWahl](#), [Programmeinstellungen](#)

1.7.1.17 Verbindungsende

Bildschirm löschen

Diese Option bewirkt, dass ZOC nach beendeter Verbindung den Bildschirm löscht und die Bildschirmfarben zurücksetzt.

Adressbuch anzeigen

Bei eingeschalteter Funktion öffnet ZOC nach dem Beenden einer Verbindung automatisch das [Adressbuch](#) (nur, wenn sich keine Einträge in der AutoWahl-Liste befinden).

Standard-Verbindungsprofil laden

Wenn diese Funktion eingeschaltet ist, lädt ZOC nach Beenden einer jeden Verbindung wieder das Standard-Verbindungsprofil.

Verbindung mit nächstem Adressbucheintrag

Wenn eine Verbindung beendet wurde und wenn sich noch ein weiterer Eintrag in der AutoWahl-Liste befindet (d.h. wenn im Adressbuch mehrere AutoWahl-Einträge zur Anwahl markiert wurden), wählt ZOC entweder automatisch oder nach gewünschter Rückfrage den nächsten AutoWahl-Eintrag der Warteschlange an.

Tab schließen

Durch Setzen dieser Option wird ein Session-Tab (Registerblatt) automatisch geschlossen wenn die Verbindung unterbrochen oder beendet wird (bezieht sich nicht auf das allerletzte Registerblatt).

Fenster schließen

Durch Setzen dieser Option wird ZOC automatisch geschlossen wenn die letzte Verbindung unterbrochen oder beendet wird.

Drücken der Eingabetaste nach Trennung verbindet erneut

Wenn diese Option aktiviert ist, wird beim Drücken der Eingabetaste in einer getrennten Sitzung versucht, die Verbindung zum Server wiederherzustellen.

Siehe auch: [Programmeinstellungen](#)

1.7.1.18 Sonstiges

Cursorposition in der Statuszeile startet mit 1/1 statt 0/0

Wenn in der Statuszeile die Cursorposition dargestellt wird (siehe [Optionen→Verbindungsprofil→Fenster](#)) legt diese Option fest, ob die linke obere Ecke als 0/0 oder 1/1 angezeigt wird.

Auswahlfenster für Rexx Dateien immer im Skript-Ordner öffnen

Wenn diese Option aktiviert ist, öffnet sich das Dateiauswahlfenster für REXX Skripte immer im Standardordner, der unter [Programmeinstellungen→Dateiablage](#) eingestellt ist. Andernfalls ist immer der Name und Ordner des zuletzt verwendeten REXX-Skripts vorgeblendet.

Siehe auch: [Programmeinstellungen](#)

1.7.1.19 *Features*

Kommunikation und Emulationen mit Priorität

In dieser Option sind die verschiedenen Verbindungstypen und Emulationen zu Gruppen zusammengefasst, in denen dem Programm mitgeteilt werden kann, ob die Funktionen häufig benötigt werden.

Die Klickfelder haben drei Zustände: Aus, An und unbestimmt. **Aus** bedeutet, dass die Funktion mit hoher Wahrscheinlichkeit nicht benötigt wird. **An** bedeutet, dass Sie ziemlich sicher benötigt wird. Der dritte Zustand steht für "vielleicht".

Je nach Einstellungen werden die zugehörigen Funktion zum Programmstart oder erst später geladen, bzw. sie werden im Dialog für die [Neue Verbindung](#) nicht angeboten, so lange Sie in einer Sitzung noch nicht anderweitig benötigt wurden.

Allerdings ist es so, dass selbst abgeschaltete Funktionen ggf. im Lauf des Programms geladen werden, wenn z.B. Einträge im [Adressbuch](#) existieren, die diese verwenden. In diesem Sinn handelt es sich hierbei nicht um eine strikte Deaktivierung sondern um Hinweise ans Programm, die den Ladevorgang optimieren.

Siehe auch: [Programmeinstellungen](#)

1.7.1.20 *Updates*

Update Einstellungen

Hier kann eingestellt werden, ob und wie oft ZOC beim Programmstart Verbindung mit der EmTec-Webseite aufnimmt und prüft, ob ein Update vorhanden ist.

Siehe auch: [Programmeinstellungen](#)

1.7.1.21 *ZOC Kennwort*

Das ZOC Hauptkennwort oder Zertifikat dient dazu, sensitive Daten (z.B. gespeicherte Kennwörter im Adressbuch oder im [Kennwortspeicher](#)) zu verschlüsseln und den Zugriff auf das Programm zu beschränken.

Hierzu kann entweder ein Kennwort festgelegt werden. Das Kennwort unterscheidet Groß- und Kleinschreibung.

Alternativ kann unter Windows ein kryptographischer Schlüssel (Chipkarte, USB-Stecker, Yubikey, etc.) nach CAC/PIV Standard verwendet werden. Es muss sich hierbei um ein Zertifikat handeln, das einen privaten RSA oder ECC Schlüssel beinhaltet, z.B. ein selbst erstellter und selbst signierter Yubikey Schlüssel. Bei Bedarf werden Sie dann aufgefordert, die entsprechende Hardware einzulegen/einzustecken.

Falls das Kennwort oder die Chipkarte/Schlüsselhardware verloren gehen, können die verschlüsselten Informationen (s.o.) nicht mehr wiederhergestellt werden.

Das Kennwort wird benötigt für ...

Hier können verschiedene Teile des Programms mit einer Kennwortabfrage versehen werden, um persönliche Informationen zu schützen.

Das Kennwort für das Programm wird nur dann abgefragt, wenn nicht bereits ein anderes ZOC Fenster geöffnet ist.

Die Option zur Kennwortabfrage, wenn für längere Zeit keine Tastatureingaben erfolgten, funktioniert auch über mehrere offene ZOC Fenster hinweg, d.h. das Programm wird nur blockiert,

wenn in keinem der ZOC Fenster eine Eingabe vorgenommen wurde. Das Entsperren eines ZOC-Fenster entsperrt automatisch auch die Kennwortabfrage anderer offener Fenster.

1.7.1.22 Sprache

Verwendete Sprache

Hier kann die in ZOC verwendete Sprache eingestellt werden. Zur Auswahl stehen Automatisch, Deutsch und Englisch. Bei Automatisch wird die Sprache von ZOC auf Basis der in Windows eingestellten Sprache gewählt.

Siehe auch: [Programmeinstellungen](#)

1.7.2 Tastaturbelegung, Zeichenumsetzung, etc.

Allgemeine Programmeinstellungen sind (im Gegensatz zu [Verbindungsprofilen](#)) für alle Verbindungen gültig. Darunter fallen Einstellungen, die sich selten verändern, nachdem sie einmal eingestellt sind.

Zu diesen allgemeinen Einstellungen gehören die folgenden:

- ☐ [Programmeinstellungen](#)
- ☐ [Symbolleiste \(Toolbar\)](#)
- ☐ [Tastenprofile](#)
- ☐ [Zeichenumsetzung](#)

Von [Tastenprofilen](#) und [Zeichenumsetzung](#) können allerdings (ähnlich wie bei den Verbindungsprofilen) mehrere Varianten abgespeichert werden, um diese dann z.B. Einträgen im Adressbuch zuzuordnen.

See also: [Optionen Überblick](#), [Optionen-Menü](#), [ZOC Konfigurieren](#), [Verbindungsprofile](#).

1.7.2.1 Tastaturprofile und Tastaturbelegung

Neubelegung von Tasten

Die Einstellungen zur Tastaturbelegung erlauben das beliebige Belegen aller Tasten auf der Tastatur mit eigenem Text, speziellen Zeichen oder ZOC-Funktionen.

Allerdings wirkt sich eine derartige Umstellung auf die Tastaturverarbeitung im gesamten Programm aus, d.h. das Umdefinieren einer Taste wie Backspace oder Alt+D überschreibt die ursprüngliche Funktionsweise dieser Taste (Backspace wird also in einer Emulation nicht mehr funktionieren und Alt+D öffnet nicht mehr das Adressbuch).

Zur Umbelegung einer Taste müssen die entsprechenden Felder für Shift, Strg, Alt, Num-Lock und Rollen je nach gewünschter Belegung ein- bzw. ausgeschaltet und danach auf die neu zu belegende Taste geklickt werden. Alternativ reicht es in den meisten Fällen aus, einfach die neu zu belegende Tastenkombination auf der Tastatur zu drücken). Im folgenden [Dialog zur Tastaturbelegung](#) kann dann die neue Belegung angegeben werden.

Scroll/Num Tastenstatus

Bitte beachten Sie, dass ZOC die Num-Lock und Rollen-Taste nicht als Haupt-, sondern als Zusatz Taste interpretiert. Das bedeutet, dass Tasten mit jeweils gedrückter oder nicht gedrückter Num-Lock oder Rollen-Taste unterschiedlich belegt werden können. Sollte dies nicht gewünscht sein, muss beim

Umbelegen der Taste das entsprechende Optionsfeld auf grau geschaltet werden (die entsprechenden Optionsfelder haben drei mögliche Zustände: ein, aus, grau).

Trotzdem ist es möglich, auch die Num-Lock und Rollen-Taste mit Text zu belegen. Allerdings verändern diese beiden Tasten dann trotz Textbelegung den entsprechenden Zustand und damit auch die LEDs auf der Tastatur. Aus diesem Grund müssen, falls diese Tasten mit Text belegt sind, alle anderen Umbelegungen so definiert sein, dass die Num-Lock und Rollen-Taste auf Zustand 'unbestimmt' gesetzt sind.

Weitere Funktionen

Der Zeigen/Drucken Knopf zeigt eine Liste aller Umbelegungen mit den zugehörigen Zusatztasten an.

Durch Verwenden der Funktionen 'Laden' und 'Speichern unter' können mehrere Tastaturbelegungen definiert und verändert werden, die dann unterschiedlichen Einträgen im Adressbuch zugeordnet werden können.

Siehe auch:

[Optionen→Verbindungsprofil→Funktionstasten](#)

[Emulationsabhängige Tastenbezeichnungen](#)

[Tasten für die VT102/VT220 Emulation](#)

[Tasten für 3270 Emulation](#)

[Tasten für 5250 Emulation](#)

1.7.2.2 Einzelne Tasten undefinieren

Nachdem unter [Optionen→Tastaturprofil](#) eine Taste zur Definition ausgewählt worden ist, erscheint ein Fenster, in dem die benutzerdefinierte Aktionen festgelegt wird, der später beim Drücken der entsprechenden Tastenkombination ausgeführt werden soll.

Einzelheiten zu den Aktionstypen und deren Funktionsweise finden Sie unter [benutzerdefinierte Aktionen](#).

Tastenbelegungen aus anderen Programmen/Quellen übernehmen: Fall Sie Werte für Tastenbelegungen in einem anderen Programm oder im Internet gefunden haben, oder wenn Sie diese per Befehl `infocmp` aus der terminfo eines Unix-Systems nachgeschlagen haben, zeigt die folgende Liste äquivalente Werte, die Sie in in ZOC für die Aktion "Text Senden" verwenden können (Beispiel siehe unten).

Andere Software	ZOC Äquivalent
^M	^M
\r	^(0D)
CR	
Return	
Esc	^[
\E	^(1B)
\e	
\033	
\x1B	
^[	^[[
	^(1B) [
CSI	^[[
	^(1B) [
	^(9B)
SS3	^[O
	^(1B) O
	^(8F)

Wenn Sie z.B. den Befehl `infocmp` auf einem Linux-System mit Xterm Emulation eingeben, sehen Sie unter anderem den Eintrag `kf12=\E[24~` (mehr dazu per `man terminfo`), der die Belegung der F12 Taste anzeigt. Der entsprechende Wert für ZOC wäre `^[ [24~` oder `^(1B) [24~`. Die komplette Liste der von ZOC unterstützten Control-Zeichen finden Sie hier: [Steuerzeichen](#).

1.7.2.3 Zeichenumsetzung

Für Benutzer in Ländern, die nicht den US-ASCII Zeichensatz verwenden, stellt sich oft das Problem, dass das von der Gegenstelle gesendete Zeichen nicht mit dem Zeichencode des Terminalprogramms oder des Betriebssystems übereinstimmt.

Hier können alle Zeichencodes für gesendete und empfangene Zeichen verändert, sowie durch Löschen aller Veränderungen der Normalzustand wiederhergestellt werden.

Durch die Funktionen Laden und Speichern unter können mehrere Zeichensatztabellen geladen oder gespeichert und diese bei Adressbucheinträgen wieder angegeben werden.

Mit dem Programm werden einige übliche Tabellen zur Umwandlung der deutschen Umlaute mitgeliefert.

Hinweis: Die Zeichenumsetzung ist nur dann aktiv, wenn die Option zum Ignorieren der Zeichenumsetzung in [Optionen→Verbindungsprofil→Terminal](#) nicht eingeschaltet ist.

Hinweis: Die Erstellung einer Umsetzungstabelle kann evtl. vermieden werden, wenn die Zeichenzuordnung bereits von ZOC durch eine vorgegebene Zeichentabelle unterstützt wird. (siehe [Verbindungsprofil→Layout](#)).

Fallbeispiel:

Angenommen, die Gegenstelle sendet das Zeichen `x` anstatt des deutschen Zeichens `Ä`. In der Codetabelle ist dieses Zeichen mit dem Code 123 zu finden. D.h. der Host sendet für ein `Ä` die 123 und ZOC müsste für die Taste `Ä` 123 senden.

Weiter angenommen, für ZOC ist der IBM/DOS Zeichensatz eingestellt (Optionen, Verbindungsprofil, Zeichensatz). In diesem Fall hat das `Ä` für ZOC jedoch den Wert 142, d.h. ein empfangenes 123 muss in 142 umgesetzt werden und ein gesendetes 142 muss in 123 verwandelt werden.

Dazu klicken Sie in der linken Tabelle auf 123 und geben als Wert 142 ein (im Fenster erscheint dann `123 x -> 142 Ä`). Dann klicken Sie in der rechten Tabelle auf 142 und geben dafür 123 an.

Hier ist eine Tabelle der Codes, die in ZOC (je nach Konfiguration) für die Umlaute verwendet werden:

Zeichen	DOS/IBM	ANSI/Windows (auch Latin-1 genannt)
Ä	142 (8E)	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 Symbolleiste (Toolbar)

Über die Symbolleiste können Funktionen als Klickfelder mit der Maus schneller erreicht werden.

Der Konfigurationsdialog für die Symbolleiste wird geöffnet indem mit der rechten Maustaste auf einem freien Feld der Symbolleiste geklickt wird. Mit der Maus können im Auswahlfenster die Klickfelder beliebig zusammengestellt werden. Die in der rechten Spalte angezeigten Einträge werden in der Symbolleiste angezeigt (allerdings nur, falls das Fenster breit genug ist).

Zusätzlich kann (je nach Bildschirmauflösung und persönlichem Geschmack) die Größe und der Abstand der Klickfelder zueinander eingestellt werden.

1.7.3 Verbindungsprofile

Verbindungsprofile sind Sammlungen von Einstellungen (z.B. Terminal, Fenster, Farben, etc.) die als Grundlage für Verbindungen dienen. Sie können mehrere solche Profile anlegen und diese dann einer oder mehreren Gegenstellen zuordnen, z.B. ein Verbindungsprofil für alle Linux-Rechner (vorausgesetzt, dass Sie für all diese die gleichen Einstellungen verwenden wollen) und ein anderes für Verbindungen zu IBM Großrechnern.

Der Dialog enthält alle Einstellungen von ZOC die verbindungsabhängig festgelegt werden können. Die Einstellungen dieses Dialogs können zusammen gespeichert und dann z.B. Einträgen im [Adressbuch](#) oder Gegenstellen im Dialog für [Neue Verbindung](#) zugeordnet werden.

Mit den Knöpfen [Speichern](#), bzw. [Speichern unter](#) können die Verbindungsprofileinstellungen in einer Datei gespeichert werden.

Ein Verbindungsprofil besteht aus:

- ☐ [Verbindungstyp](#)
- ☐ [Terminal](#)
- ☐ [Layout/Schriftart](#)
- ☐ [Farben](#)
- ☐ [Cursor](#)
- ☐ [Fensterelemente](#)
- ☐ [Emulationen](#)
- ☐ [Text Senden](#)
- ☐ [Log](#)
- ☐ [Benutzerknöpfe](#)
- ☐ [Tastatur](#)
- ☐ [Auto-Aktion](#)
- ☐ [Auto-Markierung](#)
- ☐ [Dateitransfer](#)
- ☐ [Dateibehandlung](#)
- ☐ [Inaktivität](#)
- ☐ [Trace/Debug](#)
- ☐ [Erweitert](#)

Siehe auch: [Optionen Überblick](#), [ZOC Konfigurieren](#), [Optionen-Menü](#), [Tastatur- und Zeichenumsetzungsprofile](#), [Programmeinstellungen](#).

1.7.3.1 **Verbindungstyp**

Auf dieser Seite können alle Parameter der verfügbaren und momentan ausgewählten Verbindungsmethode festgelegt werden. Der Dialog kann auch über das Optionen-Menü oder durch Klicken auf das Feld ganz links in der Statuszeile geöffnet werden.

Verbindungstyp

In dieser Liste können Standardeinstellungen für jede Kommunikationsmethode vorgenommen und ein Standardkommunikationsmittel ausgewählt werden.

Diese Angaben können aber beim Verbindungsaufbau überlagert werden, z.B. indem in einem ZOC [Adressbucheintrag](#) zwar dieses Verbindungsprofil ausgewählt, aber im Adressbucheintrag selbst dann ein anderes Kommunikationsmittel selektiert wird.

Die Angaben hier werden aber benutzt, wenn z.B. eine Verbindung über die Kommandozeile, durch ein REXX-Skript oder mittels eines Benutzerknopfes aufgebaut wird.

Der Überblick zu den einzelnen Kommunikationsmethoden ist unter [Verbindungsmethoden](#) zu finden. Details und Beschreibung der Optionen zu jeder Kommunikationsmethode finden sich in der folgenden Liste:

- ☐ Secure Shell: [Überblick](#) und [Optionen](#)
- ☐ Telnet: [Überblick](#) und [Optionen](#)
- ☐ Telnet/SSL: [Überblick](#) und [Optionen](#)
- ☐ Local Shell: [Überblick](#) und [Optionen](#)
- ☐ Serial/Modem: [Überblick](#) und [Optionen](#)
- ☐ Windows-Modems: [Überblick](#) und [Optionen](#)
- ☐ Named Pipe: [Überblick](#) und [Optionen](#)
- ☐ Rlogin: [Überblick](#) und [Optionen](#)
- ☐ UDP: [UDP \(Datagram\) Überblick](#) und [Optionen](#).

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#), [Verbindungstyp](#) und [Optionen-Menü](#)

1.7.3.2 Terminal

Auf dieser Seite können Einstellungen vorgenommen werden, die Ausgabe vom Text im Terminal.

TERMINAL-EINSTELLUNGEN

Lokales Echo

Normalerweise sendet die Gegenstelle alle ankommenden (getippten) Zeichen zum Sender zurück, weshalb sie dann für den Benutzer am Bildschirm sichtbar werden. Bei Gegenstellen die das nicht tun, kann das Terminalprogramm mit Hilfe dieser Option die geschriebenen Zeichen selbst anzeigen.

Kein extra LF bei Echo

Bei aktivierter Option (und gleichzeitig aktiviertem lokalem Echo) zeigt ZOC nur ein CR (Carriage Return) an, wenn die Return-Taste gedrückt wird (normalerweise wird als Echo ein CR und ein LF (Line Feed) angezeigt).

Beim Empfang CR -> CR/LF

Mit dieser Option wird sichergestellt, dass jedem CR ein LF folgt. Diese Funktion kann verwendet werden, wenn ein empfangener Text immer in der gleichen Zeile angezeigt wird und kein Zeilenwechsel stattfindet, oder wenn der empfangene Text wie eine Treppe nach unten formatiert ist, also zwar ein Zeilenwechsel, aber kein Sprung an den Zeilenanfang (CR) stattfindet.

Buchstaben immer Groß

Diese Einstellung kann verwendet werden, wenn für eine Verbindung nur Großbuchstaben gewünscht werden. Sie wirkt im Prinzip wie die Shift+Lock Taste an der Tastatur, wirkt sich aber nur auf Buchstaben auf, so dass andere Tasten (wie z.B. Ziffern) weiterhin normal bedient werden.

Scrollgeschwindigkeit

Graphische Anwendungen müssen zum Scrollen von Text wesentlich mehr Daten im Speicher

bewegen als textorientierte Anwendungen. Um die Textausgabe zu beschleunigen, muss also mehr als nur jeweils eine Zeile nach oben gescrollt werden; Werden z.B. immer fünf Zeilen gleichzeitig gescrollt, so wird nur noch ein Fünftel der Zeit benötigt. Je schneller die Übertragung, umso mehr Zeilen müssen gleichzeitig gescrollt werden und um so stärker beginnt das Bild im unteren Bereich zu flimmern. In den meisten Fällen genügt die Einstellung 'Mittel'.

ERWEITERT

Antwort auf ENQ

Falls die Gegenstelle eine bestimmte Antwort erwartet wenn ENQ (^E, hex 05) gesendet wird, kann hier eine Zeichenkette als Antwort eingetragen werden. Wenn die Zeichenkette die Platzhalter %USERNAME% oder %COMPUTERNAME% (oder \$(USERNAME) oder \$(COMPUTERNAME)), werden diese durch die entsprechenden Werte ersetzt.

URL-Klicks deaktivieren

Wenn diese Option aktiviert ist, wird beim Klick auf eine URL (http link, ftp link, etc.) ignoriert, d.h. kein Web Browser mit der entsprechenden Adresse geöffnet.

Sende Echo

Sendet alle empfangenen Zeichen zum Sender zurück. Diese Funktion kann verwendet werden, wenn andere Programm sich zu ZOC verbinden und erwarten, dass sich der Rechner wie ein Host-Computer verhält.

Achtung: Diese Funktion erzeugt eine Endlosschleife, wenn die Gegenstelle ebenfalls ein Echo sendet, was für die meisten Gegenstellen zutrifft.

Bit 8 ignorieren

Wenn aktiv, werden alle Zeichen im Terminalfenster als 7-Bit-Zeichen dargestellt. Dies kann Abhilfe schaffen, wenn auf dem Bildschirm anstatt Text nur Grafikzeichen erscheinen. Diese Funktion hilft aber nicht gegen falsche Zeichen, die wegen einer rauschenden Telefonleitung auf dem Bildschirm erscheinen.

Software 7E1

Manche Hosts arbeiten mit der seriellen 7E1 Zeichen Übersetzung, die normalerweise von Modems erledigt wird. In seltenen Fällen, nämlich wenn solche Host an Telnet oder ISDN Gateways hängen, kann es notwendig sein, dass ZOC die 7E1 Übersetzung durchführt. Kennzeichen für notwendiges 7E1 ist dass etwa die Hälfte der Zeichen lesbare Buchstaben und die andere Hälfte als Graphikzeichen oder europäische Umlaute erscheinen.

Piepton ignorieren

Diese Funktion legt erlaubt es, ankommende Pieptöne (^G/Bell) zu unterdrücken. Die Einstellung hier überlagert die Einstellung in [Optionen→Programmeinstellungen→Klänge](#).

Umsetzungstabellen ignorieren

Manchmal verwenden Gegenstellen einen anderen Zeichensatz, das heißt, die Gegenstelle erwartet die Darstellung anderer Zeichen für einen bestimmten gesendeten Code.

Neben den üblichen Zeichensätzen (siehe [Verbindungsprofil→Layout](#)) ist ZOC deshalb in der Lage, komplett benutzerdefinierte Umsetzungstabellen zu verarbeiten (siehe [Einstellungen→Zeichenumsetzungstabellen](#)).

Die Wirkung dieser Übersetzungstabellen kann mit dieser Funktion deaktiviert werden.

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.3 *Layout/Schriftart*

Bitte lesen Sie bzgl. dieses Dialogs auch das Thema [Zeichensätze, Fenstergröße, Farben](#).

SCHRIFTART UND SCHRIFTGRÖSSE

... Anzahl Zeilen/Zeichen im Terminal anpassen

Diese Option bewirkt, dass sich beim Ändern der Fenstergröße die Anzahl der am Bildschirm angezeigten Zeichen ändert (von z.B. 80x24 auf 87x32). Die Schriftgröße im Arbeitsbereich bleibt unverändert. Die Anzahl Zeilen/Spalten kann dann nicht manuell festgelegt werden, da sie allein von der Fenstergröße abhängt.

... Schriftgröße proportional skalieren

Diese Option bewirkt, dass beim Ändern der Fenstergröße die Anzahl der Zeilen und Spalten gleich bleibt, das Programm dafür aber die Größe der Schrift im Arbeitsbereich verändert.

Wenn diese Option aktiviert ist, kann die Größe der Schrift (siehe unten) nicht manuell eingestellt werden, da diese von der Größe des Fensters abhängt. Eine Größenänderung des Fensters mit der Maus funktioniert in diesem Fall nur in bestimmten Schrittgrößen, da nur Fenstergrößen angeboten werden, für die ein passender Zeichensatz mit natürlichen Zeichenproportionen (Verhältnis Höhe zu Breite) verfügbar ist.

... Schrift in der Höhe und Breite frei dehnen

Die Option ist ähnlich der obigen Variante mit proportionaler Skalierung, allerdings fixiert diese Option das Programm nicht auf natürliche Zeichenproportionen, sondern dehnt die Zeichenbreite, um die Fenstergröße bestmöglich zu füllen. Die erlaubt eine feinere Justierung der Fenstergröße, kann aber eine Zeichendarstellung zur Folge haben, die weniger natürlich aussieht.

... beides unverändert lassen

Diese Option bewirkt, dass Schriftgröße und Zeilen-/Zeichenzahl des Fensters erhalten bleiben. In diesem Fall kann die Größe des ZOC-Fensters nicht mit der Maus verändert werden.

TERMINALGRÖSSE

Anzahl Spalten/Zeilen

Hier kann die Größe des Übertragungsfensters (Anzahl Spalten und Zeilen) festgelegt werden. Übliche Werte sind 80x24 oder 80x25. Diese Einstellung wird von bestimmten Emulation ignoriert, wenn die Emulation aus internen Gründen einen andere fixe Größe benötigt (z.B. TN3270 oder VT100 im 132 Spalten Modus).

SCHRIFTART UND ZEICHENDARSTELLUNG

Zeichentabelle

ZOC für Windows unterstützt mehrere Zeichensatztabellen, z.B. den DOS/IBM-Zeichensatz, den Windows/ANSI Zeichensatz (auch Latin-1 genannt), sowie einige länderspezifische

Zeichentabellen und die von Linux Systemen häufig verwendete UTF8-Unicode Codierung.

Die Auswahl der richtigen Zeichentabelle hängt von der Gegenstelle ab und ist primär für die korrekte Darstellung von Umlauten und Sonderzeichen notwendig.

Schriftart

Sie können hier einen Zeichensatz aus der Liste und die entsprechende Größe auswählen. Es stehen hier nur Schriftarten zur Verfügung, bei denen alle Zeichen die gleiche Breite haben (z.B. Courier New, Lucida Console).

Die Größenangabe ist die Größe eines Zeichens in Bildpunkten (dies sind Bildschirmpixel, nicht die bei Textverarbeitungsprogrammen übliche Größenangabe "Point") in der Form **Breite x Höhe**. Aus der Wahl der Schriftgröße und der Terminalgröße (siehe [Verbindungsprofil→Fenster](#)) ergibt sich der Platz, den der Terminalbereich am Bildschirm benötigt.

Antialiasing

Mit dieser Option kann eingestellt werden, ob beim Anzeigen der Zeichen die Kanten geglättet werden sollen. Bei kleinen Zeichensätzen kann dies jedoch zu einem etwas unscharfen Bild führen, so dass die Option hier abgeschaltet werden sollte. Der dritte Zustand der Option aktiviert oder deaktiviert die Option abhängig davon was im Betriebssystem als Standard eingestellt ist.

Siehe auch: [Zeichensätze](#), [Fenstergröße](#), [Farben](#), [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.4 Terminal Farben

Dieses Fenster wird verwendet, um die Farbeinstellungen einer ZOC-Terminalsitzung auszuwählen und zu konfigurieren.

Die Farbeinstellungen sind in zwei Ebenen organisiert:

Farbauswahl: Die meisten Farben werden aus einer vordefinierten Palette ausgewählt. Zum Beispiel können Sie wählen, markierten Text in Orange aus der aktiven Palette anzuzeigen.

Palettenanpassung: Der tatsächlich verwendete Farbwert hängt von der gewählten Palette ab. Um die Farben fein abzustimmen, müssen Sie eine vorhandene Palette bearbeiten oder eine neue erstellen.

Hinweis: Wenn Sie die TN3270-Emulation verwenden, beachten Sie bitte, dass deren Farbdarstellung erheblich von den meisten anderen Emulationen abweicht. Folglich bietet TN3270 eine zusätzliche Farbkonfigurationsoption. Sie können darauf zugreifen über [Verbindungsprofil→Emulation→TN3270](#). Die TN3270 Option nutzt aber auch einige Farben aus dem Farbkonfigurationsfenster hier und verwendet auch die hier definierten Farbpaletten.

DARSTELLUNGSEIGENSCHAFTEN

Farben

Hier kann aus einer vordefinierten Farbtabelle eine Vorder- und Hintergrundfarbe angegeben werden. Zusätzlich kann Farbersatz für bestimmte Zeichenattribute gewählt werden, z.B. wenn Text in gelb angezeigt werden soll, anstatt ihn zu unterstreichen. Abhängig von der gewählten Emulation kann die Gegenstelle die Anzeigefarben (Vorder-/Hintergrund) ändern, um z.B. Text hervorzuheben.

Die Standardvorgabe von ZOC ist schwarzer Text auf weißem Hintergrund. Falls Sie hiermit mit einer Gegenstelle Probleme bekommen (z.B. weil weißer Text auf weißem Hintergrund unlesbar wird), versuchen Sie bitte die klassische Einstellung von grauem Text auf schwarzem Hintergrund.

Die ersten 16 der angebotenen Farbwerte sind Industriestandard und sollten deshalb nur mit Vorsicht verändert werden. Wenn Sie die Farbwerte trotzdem ändern möchten, kann hierzu der Knopf [Farbpalette bearbeiten](#) verwendet werden. Hiermit kann für die Grundfarben ein individueller Ersatzwert ausgewählt werden, der immer dann angezeigt wird, wenn die Gegenstelle eine bestimmte Farbe, z.B. grün anfordert.

TN3270/5250 Hinweis: Die 3270 und 5250 Emulationen benötigen nicht nur Vorder- und Hintergrundfarben, sondern zusätzlich Farben für Überschriften, Hervorhebung, Statusbereiche usw. Deshalb wurden für diese Emulationen Farbschemata (Kombinationen von aufeinander abgestimmten Farben) vorbereitet, von denen Sie eines in [Verbindungsprofil→Emulation](#) auswählen können. Diese vorgegebenen Schemata können ggf. durch die globale Farbtabelle (s.o.) modifiziert werden.

Farbpaletten

In diesem Abschnitt können Sie aus verschiedenen Farbpaletten für die Basisfarben wählen. Während die meisten Paletten auf dem gleichen Satz von Farben basieren, gibt es leichte Variationen in den Farbtönen für Farben wie Hellblau, Hellrot, Grün usw. Möglicherweise möchten Sie eine Palette wählen, an die Sie gewöhnt sind. Zusätzlich gibt es Paletten wie [Solarized](#), die nicht auf den typischen 16 ANSI-Farben basieren, sondern auf einem anderen Farbsystem.

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#), [Zeichensätze](#), [Fenstergröße](#), [Farben](#).

1.7.3.5 Cursor

CURSOR-DARSTELLUNG

Format

Form des Cursors, z.B. Block, Unterstrichen, Vertical.

Modus

Blinkend

Diese Option lässt den Cursor im Übertragungsfenster blinken.

Fadenkreuz

Hiermit wird ein grafisches Kreuz am Bildschirm angezeigt, um den Cursor besser sichtbar zu machen.

1.7.3.6 Emulationen

Hier kann die Bildschirmemulation eingestellt werden. Dieser Dialog kann über das Menü und durch Klicken auf das Emulationsklickfeld in der Statuszeile geöffnet werden.

Die hier eingestellte Emulation und Emulationsoptionen werden verwendet, so lange nicht von anderer Seite (z.B. [Adressbucheintrag](#), Dialog [Neue Verbindung](#) oder Kommandozeilenparameter /EMU) nachträglich eine andere Festlegung vorliegt.

Spezifische Tasten der einzelnen Emulation können auch auf andere PC-Tasten gelegt werden, indem die

[Tastenbezeichnungen der Emulationen](#) in Verbindung mit [Optionen→Tastaturprofilen](#) verwendet werden.

EMULATIONEN

Xterm

Die beste Wahl für den Einsatz mit Unix/Linux. Diese Emulation unterstützt Farben, nahezu alle PC-Tasten, internationale Zeichensätze und sie bietet darüber hinaus weitere moderne Features. Unter Linux wird `TERM=xterm` eingestellt.

Siehe auch [Optionen für die VT/Xterm/ANSI Emulationen](#).

ANSI BBS

Die ANSI Emulation ist die meistverwendete Emulation in Mailboxen. Der größte Vorteil ist die Verwendung des IBM-PC Grafik-Zeichensatzes und die Verwendung von Farben. Zur Kompatibilität mit dem FIDO ANSI-BBS Standard werden außerdem die meisten VT102 Kontrollcodes unterstützt. Auf Unix Systemen kann diese Emulation als `TERM=pcansi` eingestellt.

Siehe auch [Optionen für die VT/Xterm/ANSI Emulationen](#).

ANSI SCO

Eine leicht modifizierte Version von ANSI-BBS, die andere Tastencodes verwendet. Unix Systeme die sie unterstützen tun dies als `TERM=scoansi`

AVATAR/0+

Eine Emulation die fast ausschließlich im FIDO Mailbox Netzwerk verwendet wird.

TN3270

Die 3270 Emulation wird zur Ansteuerung von IBM-Großrechnern verwendet. Die Tasten sind wie unter [3270 Spezialtasten](#) beschrieben belegt. Für die verwendeten Farben kann aus verschiedenen Schemata ausgewählt werden bzw. es wird auch 3270-Extended-Color unterstützt. Das mitgelieferte Verbindungsprofil `OPTIONS\3270.ZOC` bietet außerdem einige vorgelegte Benutzerknöpfe für spezielle 3270-Funktionen.

TN5250

Die 5250 Emulation dient zur Verbindung mit IBM iSeries eServer (früher AS/400). Zur Nutzung der Emulation muss die Verbindung per Telnet aufgebaut werden (TN5250). Die 5250 Tastenbelegungen auf der PC-Tastatur ist in [5250 Tastaturbelegung](#) beschrieben. 5250 Tasten können auch auf Benutzerknöpfe oder andere PC-Tasten gelegt werden, indem die [Tastenbezeichnungen der Emulationen](#) verwendet werden.

Linux

Neben Xterm die beste Wahl für den Einsatz mit Linux. Diese Emulation unterstützt Farben und nahezu alle PC-Tasten. Unter Linux wird `TERM=linux` eingestellt.

QNX

Außerdem gibt es eine Emulation für das QNX 4.2x Echtzeit Betriebssystem. Diese Emulation wurde ursprünglich vom Ingenieurbüro Jurk in Weisenheim entwickelt und uns für die Aufnahme ins ZOC Archiv zur Verfügung gestellt. Hierfür herzlichen Dank an H. Jurk!

Sun CDE

Diese Emulation ist auf Sun Solaris (Common Desktop Environment) abgestimmt. Sie basiert auf VT220 und bietet darüber hinaus Farben. Unter Solaris wird hierzu `TERM=dtterm` konfiguriert.

TTY

Diese Emulation gibt bis auf CR, LF und TAB alle Zeichen ohne Interpretation auf dem Bildschirm aus.

VT100/VT102

VT102 wird größtenteils in Unix-Umgebungen verwendet und ist kompatibel mit dem (etwas

einfacherem) VT100-Standard. VT102 wird in 99% der Fälle funktionieren, unterstützt aber keine Farben und keine erweiterten Tasten (außer F1 - F4). Unter Unix wird hierzu `TERM=vt100` bzw. `TERM=vt102` eingestellt.

Siehe auch [Optionen für die VT/Xterm/ANSI Emulationen](#).

VT220

VT220 stellt eine erhebliche Verbesserung zu VT102 dar, und ist ein guter Startpunkt, falls Sie nicht wissen, für welche Emulation Sie sich entscheiden sollen.

VT220 verwendet die meisten Tasten, die auf einer modernen erweiterten PC-Tastatur zu finden sind, und unterstützt konsistent nationale Zeichensätze. Jedoch wird im Original keine Farbe unterstützt. Nahezu alle Unix Systeme unterstützen diese Emulation, wenn in der Shell `TERM=vt220` eingestellt wird, auch wenn die meisten Systeme heutzutage für die xterm Emulation optimiert sind.

Siehe auch [Optionen für die VT/Xterm/ANSI Emulationen](#).

VT420/VT520

Diese Terminals stellen eine weitere Verbesserung zu VT220 dar, auch wenn die meisten dieser Verbesserungen für Unix/Linux Shells nicht relevant sind. Für VAX/OpenVMS Systeme stellen diese Emulationen allerdings die erste Wahl dar.

Siehe auch [Optionen für die VT/Xterm/ANSI Emulationen](#).

VT52

VT52 ist ein sehr alter Standard von DEC. Er ist nicht aufwärts-kompatibel zu VT100, VT102 und VT220.

TERM=

Bei Telnet und SSH Verbindungen meldet ZOC die eingestellte Emulation an die Gegenstelle, die diese Information üblicherweise in der Umgebungsvariable TERM speichert. Die in den Emulationen voreingestellten Werte hierfür sind bereits auf Unix/Linux und ähnliche Systeme abgestimmt (termcaps, terminfo, etc). Falls es notwendig sein sollte, dass ZOC hierfür eine spezielle Identifikation liefert, können Sie diese hier einstellen. Üblicherweise ist dies jedoch nicht notwendig, deshalb sollte die Einstellung im Normalfall deaktiviert sein.

Siehe auch: ENQ Option in [Verbindungsprofil→Terminal](#)

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.7 Optionen für die VT/Xterm/ANSI Emulationen

Die folgenden Optionen erscheinen in einigen oder mehreren Emulationen aus der VT/Xterm/Linux/ANSI-Familie (Optionen, deren Bedeutung offensichtlich sein sollte wurden weggelassen).

VT Optionen

UTF8 (Unicode) Support aktivieren

VT und seine abgeleiteten Emulationen unterstützten ursprünglich nur 8-Bit- Zeichensätze (ähnlich wie 8859 Latin-1). Diese Option ändert das Verhalten und interpretiert 8-Bit-Zeichen als UTF8/Unicode, was heutzutage der übliche Standard für Linux-Hosts ist.

Alt-Taste als Meta-Taste verwenden

Einige Unix Applikationen (wie MC oder Emacs) verwenden Alt-Tastenkombinationen um auf spezielle Funktionen zuzugreifen. Wenn die Option aktiviert ist, fängt die Emulation Alt+Buchstabe und Alt+Ziffer ab und sendet die gedrückte Taste mit einem vorangestellten Meta-Zeichen (Esc).

Hinweis: Wenn diese Option aktiviert ist, kann nicht mit Alt-Kombinationen auf ZOC Menüfunktionen zugegriffen werden.

Navigationstasten mit Alt/Shift/Ctrl info schicken

Diese Option betrifft die Xterm Emulation und die Tasten Pos1, Ende, Einfg, Entf, Bild Auf, Bild Ab und die Cursorstasten. Wenn die Option aktiviert ist, schickt ZOC diese Tasten auch dann zur Gegenstelle, wenn Sie zusammen mit den Alt, Strg oder der Umschalt Taste gedrückt werden. Dies kann manche Tastenkombinationen des ZOC Menüs blockieren (z.B. Strg+Bild Ab für Download).

Xterm Erweiterungen deaktivieren

xterm unterstützt eine Reihe von Steuercodes, um bestimmte Operationen durchzuführen, die die nicht in der ursprünglichen VT-Emulation enthalten sind (z.B. Ändern des Programmtitels oder bestimmte Anzeigefunktionen). ZOC VT-Emulationen unterstützen diese, erlauben es aber, sie zu deaktivieren.

Farberweiterungen deaktivieren

Die ursprünglichen VT-Emulationen waren monochrom und unterstützten keine Farben. ZOC VT-Emulationen unterstützen Farben mit Steuercodes, die von den ANSI-Terminals übernommen wurden. von den ANSI-Terminals. Bei Bedarf können diese deaktiviert werden.

Rücktaste (Backspace) und Entf (Del) tauschen

Bei eingeschalteter Option wird die Entf-Taste mit der Rücktaste (Backspace)-Taste vertauscht (nur unter VT102).

Rücktaste als (Backspace)

Bei Druck der Rücktaste (Backspace) Taste wird normalerweise der Code für DEL (^?/7F) geschickt. Diese Option ändert Backspace auf ^H/08.

Oberste Reihe am Ziffernblock ist PF1-PF4

Die Originalen DEC Terminals haben die Tasten PF1 - PF4 in einer Reihe über dem Ziffernblock. Mit dieser Option ist es möglich, die Ziffernblocktasten Num, /, * und - hierfür zu verwenden.

Destruktives Backspace

Diese Option steuert, ob ein empfangenes Backspace das vorherige Zeichen löscht. Ansonsten wird nur der Cursor um ein Zeichen zurückbewegt. Emulationen (wie VTxxx) die ein eindeutig definiertes Backspace Handling haben, ignorieren diese Option.

Applikations-Modus für Nummernblock ignorieren

Im VT102-Modus wird der Ziffernblock als Ersatz für das 'VT102 Numeric Keypad' verwendet (siehe [VT102 Sondertasten](#)). Dies kann zu Problemen führen, wenn anstatt der grauen Cursorstasten die Cursorstasten im Nummernblock verwendet werden sollen. Diese Option verhindert das Umschalten des Nummernblocks in den VT102-Applikations-Modus und kann verwendet werden, wenn VT102 Applikationen vom Nummernblock keinen gesonderten Gebrauch machen.

Alternative VT220 F-Tasten

Am Markt existieren zwei verschiedene Formen von F-Tastencodes für VT220. Diese Option kann zwischen beiden umschalten (im Original VT220 Terminal wird die F5 nicht gesendet).

1.7.3.8 TN3270 Emulation

Die TN3270 verfügt über einige spezielle Einstellungen. Diejenigen, bei denen die Bedeutung nicht

offensichtlich ist, sind im Folgenden erklärt.

Optionen

Return und Strg tauschen (Senden/Zeilenwechsel)

Die normale Belegung für die Return-Taste und die rechte Strg-Taste sind Zeilenwechsel und Senden. Mit dieser Option kann die Belegung getauscht werden. (Siehe auch: [Standard-Tastenbelegung der 3270 Emulation](#)).

Fnn und PFnn Felder am Bildschirm anklickbar machen

Die Option hebt Bereiche am Bildschirm hervor, die sich auf F-Tasten beziehen (z.B. "F3=End"), und ermöglicht es Ihnen, auf diese Felder zu klicken, um die entsprechende Funktionstaste zu senden.

Shift+Pfeiltasten zum Markieren verwenden

Mit der Hochstelltaste kann Text auf dem Bildschirm markiert werden. Normalerweise erhöht die gedrückte Hochstelltaste die Cursor-Geschwindigkeit auf das Doppelte.

Cursor beim Einfügen bewegen

Diese Einstellung lässt den Cursor bewegen, wenn Text über das Clipboard eingefügt wird. Andernfalls bleibt der Cursor an der gleichen Stelle stehen.

Gleiche Inhalte können über geschützte Bereiche eingefügt werden

Wenn diese Option aktiviert ist, können Sie Text über geschützte Bereiche einfügen, wenn der eingefügte Text mit dem geschützten Inhalt übereinstimmt. Dies ermöglicht es Ihnen, einen gesamten Bildschirm mit Inhalten in den Feldern zu kopieren und später in denselben Bildschirm einzufügen (um die Felder erneut zu füllen). Wenn diese Option deaktiviert ist, springt das Einfügen beim Erreichen eines geschützten Bereichs zum nächsten Feld und fährt fort, den Inhalt dort einzufügen (ähnlich dem Verhalten von PCOMM).

SF-Query OEM-AuxDev Anfragen beantworten

Wenn ZOC eine SF-Query-Anfrage erhält, wird sie den OEM-Aux Dev 8F(hex) Bereich in die Antwort einbeziehen, welcher Informationen über den Terminal-Emulator, den Sitzungsnamen usw. enthält.

Farben

Farbeinstellungen für Standard Modus

Sie können zwischen mehreren vordefinierten Farbsets wählen, deren Farben entsprechende Bildschirm-Attribute (z.B. protected, unprotected, highlight, ...) repräsentieren. Diese Einstellungen sind nur für TN3270-Sitzungen im Standard-Modus aktiv, d.h. wenn ZOC in der blauen Statuszeile keine Kennung "-E" nach der Modellnummer anzeigt. Im erweiterten Modus bestimmt der Server die Farben, Sie können aber die Farbtöne in den ZOC 3270 Einstellungen für den erweiterten Modus anpassen (s.u.).

Farbeinstellungen für Erweiterten Modus

Dieser Dialog ermöglicht es Ihnen, eigene Farbsets für TN3270-Sitzungen im erweiterten Modus zu erstellen (ZOC zeigt dann in der blauen Statuszeile die Modellnummer die Endung "-E" an). In erweiterten TN3270-Sitzungen legt der Host eine spezifische Farbe an, um Text am Bildschirm darzustellen. Die Farbtabelle hier gibt Ihnen die Möglichkeit, die Farbwahl des Host-Rechners gegen einen anderen Farbton oder eine andere Farbe zu ersetzen. Wenn der Host also z.B. die Farbe Blau in einer TN3279-E Sitzungen anfordert, können Sie diese Farbe Hellblau oder auf die Farbe Lila ändern.

1.7.3.9 Text Senden

Diese Einstellungen werden bei Verwendung der Funktion [Senden Textdatei](#), [Senden Binärdatei](#) und beim Einfügen aus der Zwischenablage verwendet.

Pause pro Zeichen/Pause am Zeilenende

Setzt die Zeit (in Millisekunden), die nach jedem Zeichen bzw. nach jeder Zeile gewartet wird. Dies ist sinnvoll, wenn die Gegenstelle nicht in der Lage ist, die ankommenden Zeichen in voller Geschwindigkeit zu verarbeiten (z.B. in Online-Editoren).

Hinweis: Die Zeilenverzögerung wird zusätzlich zur Zeichenverzögerung gesendet. beim Versand von Binärdateien und/oder wenn die Zeichenverzögerung Null ist, wird die Zeilenverzögerung ignoriert.

Zeilenendezeichen

Hierbei handelt es sich um eine Auswahl die festlegt, mit welchem Zeichen das Ende einer Zeile an die Gegenstelle übertragen wird. Ohne Änderung überträgt das Zeilenende, wie in der Datei vorgegeben, andernfalls wird entweder CR oder LF oder beides gesendet. (Diese Option wird bei Senden Binärdatei ignoriert.)

Hinweis: Diese Funktion steht während in den Emulationen TN3270 und TN5250 nicht zur Verfügung.

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.10 *Log*

Auf dieser Seite können alle Parameter für die Protokollierung in Datei eingestellt werden. Es handelt sich um die Protokolldatei die im Log Menü und in der Statuszeile gesteuert werden kann.

Dateiname

Dieser Name wird als Dateiname für die Protokollierungsdatei verwendet. Der Name kann optional einen absoluten oder relativen Pfad enthalten (Ordner im Pfad werden bei Bedarf neu erzeugt).

Es können Platzhalter verwendet werden, um dynamische Namen für den Pfad und die Logdateien zu erzeugen, z.B. Dateinamen die das Datum oder die Bezeichnung der aktuellen Verbindung enthalten. Die Platzhalter (z.B. ^1 bis ^8) sind im Anhang bei den [Platzhaltercodes](#) beschrieben. Beispiele sind `^+_^1^2^3.log` oder `Host_^+\Log_^3_^2_^1.log`.

Hinweis: Wenn nicht explizit ein Pfad mit dem Namen angegeben ist, wird die Datei in dem Ordner gespeichert, der bei [Optionen→Programmeinstellungen→Dateiablage](#) angegeben ist.

Wenn Log-Datei bereits vorhanden ...

Wenn die zu öffnende Log-Datei bereits existiert, kann diese entweder überschrieben werden, oder die empfangenen Daten an die bereits existierende Logdatei angehängt werden. Alternativ kann dem Dateinamen eine aufsteigende Nummer angehängt werden, um sicherzustellen, dass existierende Log-Dateien nicht überschrieben werden.

Log ist aktiviert

Diese Funktion bewirkt, dass alle ankommenden Daten in der Logdatei gespeichert werden. Diese Funktion entspricht dem Logdatei-Klickfeld in der Statuszeile bzw. dem Menüpunkt [Log in Datei](#) im Log-Menü.

Neue Datei für Logfile anlegen wenn sich Platzhalter ändern

Wenn diese Option aktiviert ist und der Dateiname Platzhalter (z.B. Tag/Monat) enthält, wird eine neue Datei angelegt und in dieser weiter protokolliert, sobald sich der Platzhalter (z.B. das Datum) ändert.

Zeitmarken vor jede Zeile schreiben

Diese Einstellung schreibt im Log vor jeder Zeile die Uhrzeit damit nachvollzogen werden kann, wann genau die Zeile empfangen wurde.

Informationen über Verbindungsanfang und -ende protokollieren

Es werden bei Verbindungsbeginn in die Logdatei mehrere Zeilen mit Informationen über die angewählte Gegenstelle eingefügt (Name, Zeit des Verbindungsaufbaus, usw.).

Steuercodes für Terminalemulation nicht protokollieren

Terminalemulationen senden neben dem angezeigten Text auch unsichtbare Codes, um die Anzeige zu steuern (Farben, Cursorposition, etc.). Mit dieser Option wird festgelegt, ob die SteuerCodes im Log mitgeschrieben werden sollen oder nicht.

DC2/DC4 Steuerung

Einige Gegenstellen sind in der Lage, die Mitprotokollierung der Gegenstelle ein/auszuschalten, indem DC2 (^R) bzw. DC4 (^T) Steuerzeichen gesendet werden (das Klickfeld in der Statuszeile wird dann grau unterlegt, um anzuzeigen, dass die Mitprotokollierung in der Logdatei von der Gegenstelle unterbrochen wurde). Damit werden Meldungen der Gegenstelle (wie z.B. 'Weiter mit Return') nicht mitprotokolliert. Mit dieser Option kann die Fremdsteuerung aktiviert oder deaktiviert werden.

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.11 Fensterelemente

Hinweis: Bitte lesen Sie bzgl. dieser Einstellungen auch das Thema [Layout, Fenstergröße, Farben](#)

FENSTERELEMENTE

Menüzeile

Diese Option steuert, ob das Pulldownmenü unter der Titelleiste angezeigt wird (sämtliche Funktionen sind ggf. auch zu erreichen, wenn der Terminalbereich des Hauptfensters mit der rechten Maustaste angeklickt wird).

Symbolleiste

Hier kann die Symbolleiste ein- und ausgeschaltet werden.

Benutzerknöpfe

Hier können die Benutzerknöpfe, die unterhalb der Symbolleiste liegen, ein- und ausgeschaltet werden.

Statuszeile

Hier kann die gesamte Statuszeile ein- und ausgeschaltet werden.

LEDs in der Statuszeile

Wenn die Statuszeile aktiv ist, können dort wahlweise Statusleuchten (LEDs) aktiviert werden, die z.B. anzeigen ob Daten gesendet oder empfangen werden oder ob das Programm momentan ein REXX Skript abarbeitet oder eine andere Sonderfunktion aktiviert hat.

Ein/ausgehenden Datenverkehr anzeigen

Falls die LEDs der Statuszeile (s.o.) deaktiviert sind, wird im ersten Knopf der Statusleiste ein Symbol eingeblendet, das anzeigt, ob momentan eine Verbindung besteht oder nicht. Dieses Symbol kann auf Wunsch zusätzlich Datenverkehr anzeigen (Senden oder Empfangen von Daten auf dem Übertragungskanal).

Feld zum aktivieren/deaktivieren der Logdatei

ZOC bietet eine Funktion zum Mitschreiben des Datenverkehrs in einem Logfile. In der Statuszeile gibt es dazu ein Feld, das den Dateinamen anzeigt und zeigt, ob momentan protokolliert wird oder nicht. Hinweis: Diese Option hat drei mögliche Zustände: "An", "Aus" und "Nur sichtbar wenn Logdatei aktiv".

Cursor Position statt Anzahl Zeilen/Spalten

Mit dieser Funktion wird in einem Feld der Statuszeile die aktuelle Cursorposition statt der Bildschirmgröße angezeigt. Im gemischten Modus (halber Haken) wird die Bildschirmgröße nur angezeigt, wenn der Cursor links oben steht, andernfalls die Cursorposition. Die Anzeige der Cursorposition kann zusätzlich in [Optionen](#)→[Programmeinstellungen](#)→[Sonstiges](#) beeinflusst werden (Darstellung der Position links oben als 0/0 oder 1/1).

Rollbalken

Hier kann der Rollbalken am rechten Rand des ZOC-Fensters abgeschaltet werden (zur Wiederanzeige von Text, der inzwischen vom Bildschirm verschwunden ist).

Schnipsel

ZOC überprüft im Hintergrund alle ankommenden Daten auf Dateinamen, Internet-Adressen etc. und stellt diese in einem kleinen Fenster zusammen (siehe [Anzeigen-Menü](#)).

Mit dieser Option kann das Schnipsel-Fenster verdeckt (grau unterlegt), angezeigt oder die gesamte Funktion ausgeschaltet werden, um die CPU-Belastung zu verringern.

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.12 Benutzerknöpfe

Im Fenster unter der Symbolleiste (Toolbar) befinden sich die Leiste mit den Benutzerknöpfen, die hier bearbeitet werden können.

Benutzerknöpfe

Zu jedem Knopf kann die Beschriftung des Knopfes sowie der beim Anklicken auszuführende Funktion angegeben werden. Diese möglichen Funktionen umfassen das Senden von Text, das Starten eines REXX-Programms, den Aufruf einer Funktion aus dem ZOC Menü, usw.

Weitere Informationen hierzu finden Sie, wenn Sie einen Knopf bearbeiten und dort auf Hilfe klicken (siehe [Benutzerdefinierte Aktionen](#)).

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.13 Tastatur

F-Tasten

Es ist möglich F-Tasten mit Aktionen zu belegen, z.B. dem Senden von Text, die Auswahl einer Gegenstelle aus dem Adressbuch, der Ausführung einer Funktion aus dem ZOC Menü, oder dem Start eines REXX Scripts, usw.. Hierzu klicken Sie auf [Ändern](#) für die entsprechende Taste und legen dann die gewünschte Funktion im Fenster für [benutzerdefinierte Aktionen](#) fest.

Hinweis: Die Belegung der F-Tasten kann jedoch mit der Nutzung der F-Tasten in einer Emulation kollidieren. Falls die Tasten auch in der aktiven Emulation eine spezielle Bedeutung haben (z.B. werden von VT220 und Linux alle F-Tasten verwendet, VT100/VT102 verwenden F1-F4), wird trotzdem nur die Benutzerfunktion ausgeführt.

Falls Sie andere Tasten oder Tastenkombinationen belegen wollen, können Sie dies in [Optionen](#)→[Tastaturbelegung](#) tun.

AutoMakros

Neben den Funktionstasten bietet ZOC die sogenannten AutoMakros an. AutoMakros können mit Abkürzungen verglichen werden. Immer wenn ZOC bemerkt, dass Text getippt wird, der als Abkürzung hinterlegt ist, ersetzt ZOC diesen Text gegen den vollen Text oder führt die hinterlegte Aktion aus (siehe [Benutzerdefinierte Aktionen](#)). Bei der Textersetzung werden Backspaces gesendet, um die Abkürzung zu löschen, danach wird der volle Text gesendet.

Um zu verhindern, dass die AutoMakros fälschlicherweise ausgelöst werden, werden Groß- und Kleinschreibung unterschieden und die Abkürzung beim Tippen nur dann ersetzt, wenn ein nicht alphanumerisches Zeichen vor und hinter der Abkürzung steht. Wird z.B. `Com` als Abkürzung definiert, so wird es ersetzt, wenn `Data-Com_` oder `Com-Port` eingegeben wird, aber nicht bei `Com1-Port` oder `DataCom_`.

Es ist trotzdem sinnvoll, an jede Abkürzung z.B. einen Punkt oder ein Ausrufezeichen anzuhängen (also `MS.` und `lo!`), damit unerwünschte Aktivierung vermieden werden.

Die Funktion [Auto-Aktion](#) ist den AutoMakros sehr ähnlich und manche Ergebnisse lassen sich mit beiden Methoden realisieren. Allerdings gibt es einen entscheidenden Unterschied: AutoAktionen überprüfen den empfangenen Datenstrom (der von der Gegenstelle gesendet wurde), AutoMakros überprüfen die auf der eigenen Tastatur geschriebenen Zeichen.

Hinweis: Diese Funktion ist während der Ausführung von REXX-Programmen, während des AutoLogins, während des Lernmodus deaktiviert. Sie steht auch in den Emulationen TN3270 und TN5250 nicht zur Verfügung.

ROLLEN-TASTE

Mit dieser Option können verschiedene Funktionen auf die Scroll-Lock (Rollen) Taste der Tastatur gelegt werden.

Taste ignorieren

ZOC ignoriert den aktuellen Status der Rollen (Scroll-Lock) Taste auf der Tastatur.

Eingabezeile

Das Drücken der Scroll-Lock Taste schaltet die Eingabezeile ein und aus (siehe [Verbindungsprofil→Fenster](#) und [Bildschirmelemente](#)).

Doorway Modus

Das Drücken der Scroll-Lock Taste schaltet den Doorway Modus ein und aus. Im Doorway Modus werden DOS Tasten-Codes für Spezialtasten wie die F-Tasten, BildAuf, BildAb, usw. gesendet, die von manchen Programmen wie OS2YOU oder RemoteBBS verwendet werden.

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.14 **Auto-Aktion**

Hinweis: Die Funktionen AutoAktion steht in den Emulationen TN3270 und TN5250 nicht zur Verfügung.

AutoAktionen

AutoAktionen bieten die Möglichkeit bei Empfang einer bestimmten Zeichenkette von der Gegenstelle, einen Text zurückzusenden oder andere ZOC-Aktionen auszuführen (siehe [Benutzerdefinierte Aktionen](#)). Hierzu muss lediglich der Text, der die Aktion auslösen soll, sowie die Aktion selbst bestimmt werden. Diese Funktion kann z.B. verwendet werden für

- ☐ Automatische Logins: Beim Empfang von `Benutzer?` wird automatisch der Benutzername und das Kennwort gesendet.
- ☐ Nachrichten lesen: Nach Empfang von `You have mail` kann das entsprechende Nachrichtenprogramm automatisch geladen werden (z.B. `elm^M`).

Die Funktion AutoAktion ist den [AutoMakros](#) sehr ähnlich und manche Ergebnisse lassen sich mit beiden Methoden realisieren. Allerdings gibt es einen entscheidenden Unterschied: AutoAktionen überprüfen den empfangenen Datenstrom (der von der Gegenstelle gesendet wurde), AutoMakros überprüfen die auf der eigenen Tastatur geschriebenen Zeichen.

Achtung: Es sollte sichergestellt werden, dass der als Aktion angegebene Text eindeutig ist. Es ist sicherer, auf den Text `--Weiter--` zu warten als nur auf den Text `Weiter`.

Hinweis: Diese Funktion ist während der Ausführung von REXX-Programmen, während des AutoLogins und während des Lernmodus ausgeschaltet.

Siehe auch: [Verbindungsprofile](#), [Benutzerdefinierte Aktionen](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.15 **Auto-Markierung**

Die Funktion Auto-Markierung erlaubt es, spezifischen Text (z.B. eine wichtige Systemmeldung) farblich hervorzuheben.

Normaler Text:

Die Einstellungen im Verbindungsprofil erlauben es, solche Hervorhebungen anzulegen, zu ändern oder zu löschen. Falls dort für eine Farbe 'Standard' ausgewählt wird, bedeutet dies, dass die Vorder- oder Hintergrundfarbe angezeigt wird, die für den Text normalerweise verwendet worden wäre.

Die Texte werden beim Empfang von der Gegenstelle exakt so verglichen wie angegeben, außer die Optionen zum Ignorieren von Groß/Kleinschreibung oder zur Verwendung von Platzhaltern werden gesetzt.

Platzhalter:

Bei der Verwendung von Platzhaltern kann ein zu suchender Text spezielle Zeichen enthalten, die nicht das Zeichen selbst repräsentieren, sondern stattdessen eine bestimmte Klasse von Zeichen: Das `#` Zeichen repräsentiert eine Ziffer (0-9), das `@` Zeichen einen Klein- oder Großbuchstaben, das `?` stimmt mit jedem Zeichen überein, und ein `%` repräsentiert ein beliebiges Zeichen, das weder ein Buchstabe noch eine Zahl ist. So wird z.B. der Platzhalter `System Error (@####)` immer dann ausgelöst werden, wenn der Text "System Error" gefolgt von einem Leerzeichen und einem Fehlercode in Klammern, der aus einem Buchstaben und vier Ziffern besteht.

Hinweis: Jeder Platzhalter bezieht sich auf genau ein Zeichen (d.h. es existiert kein Äquivalent zum `*` Platzhalter, wie er häufig in Dateinamen und regexp-Ausdrücken verwendet wird). Weiterhin gilt die Einschränkung, dass das letzte Zeichen eines zu suchenden Textes kein Platzhalter sein darf.

Regular Expressions:

Hinweis: Diese Funktion ist noch in einem experimentellen Stadium.

Der Suchtext kann auch aus einer einfachen Regular Expression bestehen. Diese Funktionalität ist sehr CPU-hungrig, bitte limitieren Sie die Verwendung von Regular Expressions, wenn Sie große Mengen an Textoutput erwarten, wie z.B. beim Befehl `ls -R` in einer Linux-Shell.

Benötigt wird eine Regular Expression (siehe unten), zusammen mit einer Begrenzungs-Typklasse und der maximalen Länge an Text, gegen den die Regular Expression angewendet wird.

Es wird erwartet, dass die Zeichen der angegebenen Begrenzungs-Typklasse das potentielle Ergebnis begrenzen. Wenn Sie z.B. Whitespace-Zeichen als Begrenzungs-Typklasse angeben, dann wird die Regular Expression nur dann einen Treffer landen, wenn beide Seiten des Ausdruckes von einem Whitespace begrenzt werden (der Beginn und das Ende einer Zeile gelten für alle Klassen als mögliche Begrenzung). Stellen Sie sich zum einfacheren Verständnis vor, dass ein einzelnes Zeichen dieser Klasse an den linken und rechten Rand der Regular Expression hinzugefügt wird. Ein Suchmuster von z.B. `\d+` mit einer Whitespace-Begrenzungs-Typklasse wird die Zahl 999 in `a = 999 + b;` als Treffer ansehen, nicht aber bei `a= 999+b;`.

Die maximale Länge gibt an, welche maximale Länge von Ihnen als Treffer erwartet wird. Längere Zeichenketten, selbst wenn diese zwischen den zu erwartenden Begrenzungs-Typklassen-Zeichen stehen, werden nicht als Treffer angesehen. So wird z.B. ein Suchmuster von `\d+` mit einer Begrenzungs-Typklasse Non-Digits und einer maximalen Länge von 8 zwar den Text in `a= 123;` als Treffer finden, nicht aber `a= 333666999;`.

Regular Expression Elemente:

Die regex Suche unterstützt nur die häufigsten grundlegenden Regular Expression Typen.

- `.` Punkt, passt für jedes Zeichen
- `*` Stern, passt für kein Zeichen oder mehr (greedy)
- `+` Plus, passt für ein Zeichen oder mehr (greedy)
- `?` Fragezeichen, passt für kein Zeichen oder ein Zeichen (non-greedy)

`[abc]` Zeichenklasse, passt für ein Zeichen aus {'a', 'b', 'c'}

`^[abc]` Invertierte Zeichenklasse, passt wenn NICHT ein Zeichen aus {'a', 'b', 'c'}

`[a-zA-Z]` Zeichenbereiche, passt für jedes Zeichen innerhalb des Bereichs a-z oder A-Z

`\` Exakt das folgende Zeichen, z.B. `\*` passt für ein *-Zeichen. Ausnahme sind die Zeichen `aAsSwWdD` (siehe unten), bei denen die Kombination für ganze Gruppen von Zeichen steht.

`\s` Whitespace und alle Steuerzeichen (Hex 00 - 1F)

`\S` Nicht-Whitespace

`\w` Alphanumerisch/Worte `[a-zA-Z0-9_]`

`\W` Nicht-Alphanumerisch

`\d` Digits, `[0-9]`

`\D` Nicht-Digits

`\a` Alphabetische Zeichen, Latin-Zeichen `[a-zA-Z]`

`\A` Nicht-Alphabetische Zeichen

`^` Nicht unterstützt

`$` Nicht unterstützt

`(a|b)` Klammern und Oder-Ausdrücke werden nicht unterstützt

`{m,n}` Geschweifte Klammern mit Aufzählungen werden nicht unterstützt

Regular Expression Beispiele:

`\d+\.\d+\.\d+\.\d+` Etwas, was grob einer IP-Adresse entspricht (unter Verwendung der Non-Digit

Begrenzungsklasse).

\a\a\a\a\d\d\d\d Ein Fehlercode mit vier Zeichen und vier Ziffern (unter Verwendung der Non-Alnum Begrenzungsklasse).

1.7.3.16 Dateitransfer

Hier können alle Parameter für die Dateiübertragung eingestellt werden. Diese Seite kann auch durch Klicken auf das Protokoll-Klickfeld in der Statuszeile geöffnet werden.

PROTOKOLL

Es stehen X- Y-, Zmodem und einige andere Protokoll zur Auswahl. Zmodem ist davon das komfortabelste und das am weitesten verbreitetste Protokoll.

ZMODEM

Zmodem ist eines der meistbenutzten Dateiübertragungsprotokolle. Es erlaubt den Austausch von Dateien über nahezu jede Art von Verbindung und ist für Übertragung mehrerer Dateien, Erhalt von Dateiattributen und Korrektur von Fehlern bei schlechten Leitungen sind Kennzeichen von Zmodem. Wo immer es verfügbar ist, sollte die Verwendung von Zmodem dringend erwogen werden.

Siehe [Zmodem Dateiübertragung](#) für die Beschreibung der Installation und Verwendung Zmodem bzw. Liste der Zmodem Optionen.

SCP

Für Filetransfer via SCP gibt es keine weiteren Einstellungen. Wenn SCP ausgewählt ist und Sie mit einem Server via SSH Verbunden sind, wählen Sie aus dem Transfermenü die Punkte [Upload](#) oder [Download](#) um eine Datei zu übertragen (siehe auch [SCP Dateitransfer](#)).

KERMIT

Kermit ist ein Protokoll, dessen Vorteile in der weiten Verbreitung auf nahezu allen Plattformen und guten Zuverlässigkeit bei schlechten Verbindungen liegen.

Im Normalfall sollten die Standardwerte der Einstellungen, die mit dem Resetknopf wiederhergestellt werden können, funktionieren. Die verwendete Blockgröße kann zwischen 80 und 9024 Zeichen betragen, auf 7-bit-Leitungen muss die 8-Bit-Einstellung abgeschaltet werden.

IND\$FILE

Das IND\$FILE Dateiübertragungsprotokoll wird in mit TN3270 Verbindungen verwendet, um Dateien mit IBM Großrechnern auszutauschen (siehe [IND\\$FILE Transfer](#) für weitere Details).

YMODEM

Einige Systeme bieten fälschlicherweise die Begriffe Ymodem für Xmodem mit Block-1024 und Ymodem Batch für normales Ymodem an. In diesen Fällen wählen Sie in ZOC statt Ymodem das Protokoll X-Modem und aktivieren die Option "1KB Blöcke", bzw. wählen Ymodem (wenn Ymodem-Batch verlangt wird).

XMODEM

Diese Einstellungen sind nur bei aktiver Xmodem-Einstellung wirksam.

CRC ein

CRC wird vom Empfänger gesteuert. Es werden 16-Bit CRC-Prüfsummen anstatt der etwas fehleranfälligeren 8-Bit Xmodem-Prüfsummen verwendet.

1KB Blöcke

Mit dieser Funktion verwendet ZOC Blocklängen von 1024 Zeichen anstelle von 128. Einige Systeme nennen diese Option fälschlicherweise Ymodem.

Chop ein

Wenn aktiv, versucht ZOC, überflüssige Füllzeichen am Ende der übertragenen Datei zu entfernen.

EIGENSCHAFTEN/SONSTIGES

Übertragung wenn möglich automatisch erkennen und starten

Mit dieser Option startet ZOC einen Datei-Transfer automatisch, falls die Gegenseite eine entsprechend eindeutige Zeichenfolge schickt (dies ist nicht bei allen Arten von Transfers möglich und betrifft vor allem Zmodem oder Kermit Transfers).

Übertragungsfenster nach dem Transfer sofort schließen

Mit dieser Option wird die Wartezeit von 3 Sek. nach einem Dateitransfer abgeschaltet.

Zusammenfassung des Transfer am Bildschirm unterdrücken

Wenn Diese Option aktiviert ist, wird außerhalb des Dateitransfer-Fensters keine Liste/Zusammenfassung der übertragenen Dateien angezeigt (d.h. der Textbildschirm der Session bleibt unverändert).

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.17 Dateibehandlung

Hier können alle Parameter für die Dateibehandlung eingestellt werden.

WENN DOWNLOAD-DATEI BEREITS EXISTIERT ...

Hier kann angegeben werden, was geschehen soll, wenn während eines Downloads eine Datei empfangen wird, die im Download-Verzeichnis schon existiert. In diesem Fall können verschiedenste Aktionen angegeben werden, was z.B. passieren soll, wenn die gerade empfangene Datei ein älteres Datum, gleiches Datum oder neueres Datum hat. Zusätzlich kann festgelegt werden, ob gegebenenfalls die alte oder die neue Datei umbenannt werden soll.

SPEZIELLE DATEIENDUNGEN

ZOC ist in der Lage, Dateien mit bestimmten Endungen in einem alternativen Verzeichnis abzuspeichern (siehe [Optionen→Programmeinstellungen→Dateiablage→Alt. Verz.](#)) bzw. Dateien mit bestimmten Endungen nach erfolgreichem Upload zu löschen.

Die angegebenen Dateiendungen können Platzhalter wie '*' oder '?' enthalten. Mehrere Endungen müssen mit '|' getrennt werden (ohne Leerzeichen), also z.B. `GIF|Q*|MO?|TU?`

Hinweis: Es dürfen nur Dateiendungen angegeben werden, keine kompletten Dateinamen (nicht '*.GIF|ABC.DOC').

Hinweis: Zur Realisierung eines komplexeren Steuersystems zur Verwaltung von Downloaddateien, kann die REXX-Datei für [ZOC Ereignisse](#) (siehe [Optionen→Programmeinstellungen→Spez. Dateien](#)) verwendet werden).

SONSTIGES

JPG/GIF-Bilder während des Downloads anzeigen

ZOC ist in der Lage, Bilder im GIF- und JPG-Format gleich während des Downloads anzuzeigen. Hierzu befindet sich im ZOC-Verzeichnis die Datei SHOWEM.DLL. Diese Funktion kann, wenn nicht gewünscht, abgeschaltet werden.

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.18 Inaktivität

INAKTIVITÄT

Hier kann eine Aktion festgelegt werden, die ausgeführt wird, wenn eine bestimmte Zeit lang weder Zeichen gesendet noch empfangen wurden.

Falls man bei Inaktivität einen Verbindungsabbruch durch die Gegenstelle verhindern möchte, ist es möglich in diesem Fall automatisch Zeichen zu senden oder eine andere benutzerdefinierte Aktion auszuführen (z.B. eine Skript zu starten oder eine Sondertaste einer Emulation zu senden).

Bei Senden von Zeichen sollte zunächst versucht werden, Zeichen zu verwenden, die keine ungewollten Aktionen bei der Gegenstelle auslösen, z.B. das Nullzeichen (hex 00), Esc oder notfalls Enter. Für Telnet-Verbindungen gibt es hierzu auch das TELNET_NOP Paket.

Alternativ kann ZOC nach Inaktivität die Verbindung unterbrechen, so dass z.B. Modem-Verbindungen automatisch in der Dauer begrenzt werden.

Um bei einem Timeout keine Aktion auszuführen, muss die Timeoutzeit auf Null gesetzt werden.

Hinweis: Für SSH und Telnet Verbindungen stehen für diesen Zweck zusätzlich spezielle Optionen zur Verfügung (Keep-Alive). Sie finden diese in den jeweiligen Einstellungen für Telnet- bzw. SSH. Siehe hierzu auch das [Tutorial: Configure Connection Type Options \(englisch\)](#).

1.7.3.19 Trace/Debug

Hier finden sich Einstellungen, um bitgenaue Protokolle der gesendeten und empfangenen Daten zu erzeugen. Die folgenden Funktionen werden deshalb normalerweise nur von Entwicklern und Technikern verwendet, um den Datenstrom zwischen Gegenstelle und eigenem Gerät in bitgenauer Form sichtbar zu machen.

Versteckte Steuerzeichen im Terminal anzeigen statt ausführen

Empfangenen Steuerzeichen (hex 00 bis 1F) werden sichtbar am Bildschirm angezeigt, anstatt die damit assoziierten Funktionen auszuführen (z.B. `^I` anstelle von Tab).

Ausgabe als Hex-Dump im Terminal

Zeigt alle empfangenen Zeichen als hexadezimale Werte am Bildschirm an.

Daten als Hex-Dump ins Datenprotokollfenster schreiben

Zeigt alle empfangenen Daten als Hexdump im Datenprotokollfenster statt im Terminal-Fenster an.

Daten als Ascii-Trace ins Datenprotokollfenster schreiben

Zeigt alle empfangenen Daten als Ascii-Trace im Datenprotokollfenster statt im Terminal-Fenster an.

Ascii Trace in Datei schreiben

Schreibt alle von ZOC gesendeten und empfangenen Zeichen als Text (zum Teil HEX) in die Datei `zoctrace.txt`. Aus der Datei sind Zeitpunkt des Sendens/Empfangs und alle Text-, Steuer- und Sonderzeichen ersichtlich.

Die Datei wird im Ordner für Logdateien gespeichert, der über das Log-Menü geöffnet werden kann.

Binär Trace in Dateien schreiben

Schreibt alle empfangenen Zeichen in Rohform in die Datei `zoctrain.bin` und alle gesendeten Zeichen in die Datei `zoctrout.bin`.

Beide werden im Ordner für Logdateien gespeichert, der über das Log-Menü geöffnet werden kann.

Low-level Kommunikations Trace-Dateien schreiben

Auch diese Option schreibt zwei Trace-Dateien in Binärformat. Abhängig von der verwendeten Verbindungsmethode enthalten diese Dateien aber außerdem noch Steuerzeichen, die intern vom Kommunikationsdevice verwendet werden. Für Telnet-Verbindung enthält die Low-level Trace-Datei z.B. auch Telnet-Protokoll-Negotiate-Sequenzen wie IAC Sequenzen, End-Of-Record Markierungen etc. Beide werden im Ordner für Logdateien gespeichert, der über das Log-Menü geöffnet werden kann.

Support Debug-Protokoll

Dieses Protokoll enthält Daten, die nur für den EmTec Support wichtig sein können. Aktivieren Sie diese Option nur, wenn Sie vom EmTec Support dazu aufgefordert werden.

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.7.3.20 Erweitert

Erweiterte Emulationssequenzen zur Ausführung lokaler Befehle zulassen

Diese Option aktiviert diejenigen der [erweiterten Emulationssequenzen](#), die normalerweise deshalb deaktiviert sind, weil sie die Ausführung von Befehlen auf dem lokalen Computer erlauben und weil dies ein massives Sicherheitsproblem darstellen kann. Deshalb sollte diese Option auch nur aktiviert werden, wenn der Gegenstelle mit der man verbunden ist *absolut* vertraut wird.

Wenn Dateien über das Terminal gezogen werden

Über das Terminalfenster gezogene Dateien können entweder per Dateitransfer übertragen werden (falls der eingestellte Dateitransfer keinen Mehrdateien-Transfer unterstützt, wird nur die erste Datei übertragen), oder die Dateinamen dieser Dateien an die Gegenstelle übertragen werden.

1.7.3.21 Überblick über Farben/Zeichensätze/Fenstergröße

Da sich eine Terminalemulation in der Behandlung von Text von einer Textverarbeitung unterscheidet, finden Sie im Folgenden Erklärungen zu Zeichensätzen, Farben etc.

Zeichensätze

Zur Nutzung in einer Terminalemulation kommen nur Zeichensätze in Frage, bei denen alle Zeichen die gleiche Breite aufweisen, d.h. bei denen ein `i` genauso breit ist wie z.B. ein `w`. Dies ist notwendig um z.B. Tabellen korrekt darzustellen. Unter Windows trifft dies z.B. auf die Zeichensätze Lucida Console oder Courier New zu.

Weiterhin ist es je nach Gegenstelle notwendig, dass die Zeichensätze bestimmte Zeichen darstellen können, die z.B. zur Darstellung von Linien oder Kästen notwendig sind. Die Codierung dieser Zeichen ist über sog. Charactersets (Zeichentabellen) geregelt, die gebräuchlichsten sind ANSI/Windows/Latin (enthält europäische Umlaute, jedoch keine Grafikzeichen) und IBM/DOS (enthält Grafikzeichen, jedoch nicht alle Umlaute).

Die Einstellung erfolgt unter [Verbindungsprofil→Layout](#)

Fenstergröße

Die Anzahl der darzustellenden Zeichen hängt auch von der Gegenstelle ab, üblich sind z.B. 25 Zeilen zu je 80 Zeichen. Da alle Zeichen einer Schriftart eine fixe Höhe und Breite haben, ergibt sich daraus ein Bereich fester Größe, der für die Darstellung benötigt wird. 80x25 Zeichen dargestellt mit kleinen Zeichen benötigen entsprechend weniger Platz als mit großen. Da der Darstellungsbereich fix ist, kann das ZOC Fenster nicht beliebig vergrößert oder verkleinert werden. ZOC bietet verschiedene Einstellungen, wie mit der Größe des Darstellungsbereichs verfahren werden soll, wenn die Größe des ZOC-Fensters mit der Maus geändert wird, diese unterliegen aus den o.g. Gründen jedoch gewissen Einschränkungen. Die Auswahl der Anzahl Zeilen/Zeichen erfolgt unter [Verbindungsprofil→Terminal](#), die des Verfahrens bei Größenänderung unter [Verbindungsprofil→Layout](#)

Farben

Die meisten Emulationen basieren auf einer Palette von 16 Grundfarben. Unter [Verbindungsprofil→Fenster](#) können Sie die Standardfarben für Text, bzw. Hintergrund auswählen. Die Gegenstellen können jedoch mit Hilfe der eingestellten Emulation die Darstellung auf bestimmte Farben umstellen (z.B. Standard-Gelb für Hervorhebung), diese harmonisieren dann mehr oder minder mit den selbst gewählten Farben, da die Gegenstellen üblicherweise von hellgrauem Text auf schwarzem Grund ausgehen.

Im den o.g. Fenstereinstellungen ist es möglich die 16 Grundfarben für ZOC zu verändern, das heißt es kann z.B. ein anderer Farbton für hell- und dunkelgrün gewählt werden, bzw. im Extremfall grün durch eine andere Farbe ersetzt werden, die immer dann dargestellt wird, wenn die Gegenstelle Darstellung in grün anfordert.

Die Emulationen 3270 und 5250 haben eigene Farbtabelle, hier wird von der Gegenstelle ein Farbtyp angefordert, z.B. Standard-Textfarbe, Farbe für Überschriften usw. Die wirklich darzustellenden Farben (z.B. Textfarbe grün, Überschriftsfarbe türkis, usw.) sind in ZOC in Farbschemata hinterlegt, deren Farben hinsichtlich guter Lesbarkeit aufeinander abgestimmt sind. Eins dieser Farbschemata kann unter [Verbindungsprofil→Emulation](#) jeweils für die 3270 und 5250 Emulation ausgewählt werden. Die Einzelfarben können ggf. über den oben erwähnten Dialog zur Änderung der globalen Farbtabelle modifiziert werden, da die Schemata auch aus den 16 Grundfarben zusammengestellt sind.

Siehe auch: [Verbindungsprofile](#), [ZOC Konfigurieren](#) und [Optionen-Menü](#)

1.8 Verbindungsmethoden

ZOC unterstützt eine Reihe von Verbindungsmethoden, über die kommuniziert werden kann. Nachfolgend eine Liste der Verbindungstypen und deren Verwendung, Einstellungen usw.

- ☐ [Secure Shell \(SSH\)](#)
- ☐ [Telnet](#)
- ☐ [Telnet/SSL](#)
- ☐ [Serial/Direct und Serial/Modem](#)
- ☐ [Windows-Modems \(TAPI\)](#)
- ☐ [Local Shell \(Windows Command Prompt\)](#)
- ☐ [Named Pipe](#)
- ☐ [UDP \(Datagram\)](#)
- ☐ [RLogin](#)

1.8.1 Secure Shell (SSH)

Secure Shell ist eine Verbindungsmethode, die eine verschlüsselte Verbindung über unsichere Netzwerke ermöglicht.

Die folgenden Themen behandeln verschiedene Aspekte der Verwendung von SSH in ZOC:

- ☐ [SSH Überblick](#)
- ☐ [SSH File Transfer](#)
- ☐ [SSH Einstellungen](#)
- ☐ [Globale Einstellungen](#)
- ☐ [Menüfunktionen](#)
- ☐ [REXX ZocDeviceControl](#)
- ☐ [Erzeugen von Public/Private-Keys](#)
- ☐ [Ändern von Passphrases](#)
- ☐ [Tunnelprofile erstellen](#)

1.8.1.1 *Secure Shell (SSH) Überblick*

Der Verbindungstyp Secure Shell erlaubt, sich an einer Gegenstelle im Internet/Intranet anzumelden. Im Gegensatz zu Telnet verwendet SSH eine Datenverschlüsselung für alle gesendeten und empfangenen Daten, was es Dritten unmöglich macht, über die verwendete Leitung Kennwörter oder vertrauliche Daten erhalten zu können.

Hinweis: Die Implementierung von SSH in ZOC basiert auf dem SSH-Client der OpenSSH-Distribution. Weitere Informationen zu OpenSSH finden Sie unter <http://www.openssh.org>.

Login

SSH benötigt die Angabe von Benutzer und Kennwort bzw. Authentifizierungsdatei noch vor dem Aufbau der Verbindung. Bei Anwahl mit Hilfe des Adressbuchs werden diese Daten aus den entsprechenden Feldern im Adressbucheintrag verwendet. Bei einer manuellen Anwahl müssen diese Daten im Verbindungsdialog angegeben werden, bevor der OK-Knopf gedrückt wird. Bzgl. Anmeldungen mit kryptografischen Schlüsseldateien lesen Sie bitte auch [Erzeugen von Public/Private-Keys](#).

File Transfer

Informationen zur Übertragung von Dateien über eine SSH Verbindung auf einen anderen Rechner finden Sie unter [SSH File Transfer](#).

1.8.1.2 SSH File Transfers

Um Dateien über SSH auszutauschen, verwenden Sie am besten SCP oder Zmodem. Dazu müssen Sie eines von beiden als Dateitransferprotokoll unter [Optionen](#)→[Verbindungsprofil](#)→[Transfer](#) einstellen. Falls die Verbindung über das ZOC Adressbuch aufgebaut wird, können Sie SCP oder Zmodem auch in den Adressbucheinträgen im Register Optionen auswählen.

SCP

Senden und Empfangen von Dateien wird über die Befehle Upload und Download im Transfer-Menü von ZOC angestoßen, während Sie am fremden Rechner in einer Shell stehen (der `scp` Befehl muss dort verfügbar sein).

Die Dateien am Server beziehen sich auf das Arbeitsverzeichnis der Shell (`pwd`), in ZOC sind die lokalen Ordner unter [Optionen](#)→[Programmeinstellungen](#)→[Dateiablage](#) konfiguriert.

Siehe auch: [SCP Dateitransfer](#)

Zmodem

Die meisten Unix/Linux Server verfügen über Zmodem Programme (rz und sz), die Dateitransfer von und zu entfernten Rechnern mit ZOC auf elegante Weise ermöglicht. (Falls Zmodem auf einem Server nicht installiert ist, suchen Sie im Internet nach lrzsz.zip, das dann auf die unter Unix übliche Art und Weise (./configure, make, make install) installiert werden kann.

Siehe [Zmodem Dateiübertragung](#) für die Beschreibung der Verwendung Zmodem auf Unix/Linux Rechnern, bzw. für eine Liste der Optionen.

1.8.1.3 Secure Shell (SSH) Optionen

Um SSH-Verbindungsoptionen anzupassen, navigieren Sie zum Abschnitt Verbindungstyp im [Verbindungsprofil](#).

Diese Änderungen gelten dann für alle Verbindungen, die dieses Verbindungsprofil verwenden. Wenn Sie die Optionen für eine einzelne Verbindung ändern möchten, klicken Sie auf die Schaltfläche [Konfigurieren](#) im [Neue Verbindung](#)-Fenster.

Für Verbindungen, die über das [Adressbuch](#) hergestellt werden, verwenden Sie die Schaltfläche [Konfigurieren](#) im [Host-Tab](#) Ihres Adressbuch-Eintrags.

Nachfolgende Optionen sind für SSH Verbindungen verfügbar:

Connect via Proxy

Falls die Verbindung zum SSH Server nicht direkt sondern über den Umweg über einen Proxy aufgebaut werden soll, können Sie in den SSH Optionen im Feld Proxy den Proxy-Typ und den Namen oder IP des Proxy Servers mit optionalem Port (Standardport ist 1080) angeben, z.B.

`192.168.1.1` oder `myproxy.somewhere.com:8080`

Wenn der Proxy vom Typ SOCKS5 oder HTTP ist, können außerdem ein Benutzername und ein Passwort in der Form `user:pass@host:port` übergeben werden.

Ein [Jumpserver](#) (bzw. [Bastion-host](#) ist eine SSH-Gegenstelle, die als Gateway zum tatsächlichen Ziel-Host fungiert. Technisch gesehen wird erst die Verbindung zum Jumpserver hergestellt, und nach erfolgter Authentifizierung am Jumpserver dort ein SSH-Befehl ausgeführt, der dann die Verbindung zum tatsächlichen Zielsystem herstellt. Falls der Login auf dem Jumpserver einen anderen Benutzernamen hat, können Sie ihn in der Form `user@host:port` angeben.

Hostkey mit 'known_hosts' abgleichen

Weiterhin erlaubt die SSH-Unterstützung von ZOC das Speichern und Überprüfen des von der Gegenstelle gesendeten Hostkeys. Bereits gespeicherte Hostkeys befinden sich in den Dateien `known_hosts` für SSH1 und `known_hosts2` für SSH2-Verbindungen im Unterverzeichnis 'ssh'. Der Dateiaufbau dieser Dateien ist identisch mit dem Aufbau der unter Unix verwendeten OpenSSH Anwendung.

Keep-Alive Signal aktivieren

Die Keep-Alive Option ist nützlich wenn die Gegenstelle die Verbindung bei Inaktivität des Nutzers unterbricht. Wenn die Option gesetzt wird, werden sowohl die IP Verbindung auf Keep-Alive gesetzt (SO_KEEPALIVE, bzw. äquivalent `keepalives` im OpenSSH) als auch bei SSH 2.0 auf Applikationslevel alle 60 Sekunden Keep-Alive Pakete geschickt (entspricht `ServerAliveInterval=60` im OpenSSH).

Kennwort-Abfragen im Terminal anzeigen

Wenn der Server ein Passwort oder eine Passphrase von ZOC abfragt, wird normalerweise ein Dialogfenster geöffnet, um diese Daten abzufragen. Ist diese Option aktiviert, werden die Abfragen stattdessen als Text im Terminal-Fenster abgefragt, ähnlich wie beim Befehl `ssh` von OpenSSH.

Globale Schlüsseldateien bearbeiten

Zusätzlich können über die Schaltfläche "Globale Schlüsseldateien bearbeiten" für SSH-Verbindungen bis zu drei Authentifizierungsdateien angegeben werden, die dann von ZOC für die Authentifizierung verwendet werden können. Die Dialoge jedes Adressbucheintrages sowie der manuellen Anwahl bieten die Möglichkeit, für die entsprechende Verbindung entweder die globalen Authentifizierungsdateien zu verwenden, oder aber für diese Verbindung eine spezielle Datei zu verwenden. Siehe auch: [SSH→Globale Schlüsseldateien](#)

ERWEITERTE OPTIONEN

Authentifizierungs-Methoden

Alle verfügbaren Authentifizierungs-Methoden können aktiviert oder deaktiviert werden. ZOC-Terminal und der SSH-Server werden versuchen, gemeinsame Methoden zu finden, und prüfen, ob sich der Benutzer über eine dieser Methoden authentifizieren kann. Die Reihenfolge, in der die Methoden (falls aktiviert) abgefragt werden, ist: gssapi, publickey, keyboard-interactive, password. Sollte eine dieser Methoden als "Bevorzugt" markiert sein, wird diese auf den ersten Platz dieser Reihenfolge gesetzt.

Public-Key Authentifizierung

Aktiviert oder deaktiviert die Public-Key Authentifizierungs-Methode. Diese Methode basiert auf Public-Private-Schlüsseldatei-Paaren, um die eigenen Identität zu bestätigen.

Agent Support

Wenn eine dieser Einstellungen aktiviert ist, wird beim Login bzw. bei späteren SSH-Verbindungen innerhalb der Sitzung ein Agent kontaktiert.

In der Option `intern` wird hierbei von ZOC selbst nur der Private-Key, der für den Login am Host verwendet wurde (falls überhaupt Private-Key Login stattfand), dafür verwendet, um weitere SSH-Befehle, die innerhalb der aktuellen Sitzung ausgehen, zu authentifizieren. Dies entspricht (eingeschränkt) der Funktionalität, die der OpenSSH-Parameter `-A` in Kombination mit `ssh-agent` anbietet.

Bei den anderen Möglichkeiten wird entweder der Putty-Agent (`pageant.exe`) oder der OpenSSH `ssh-agent` oder der `zoc-agent` (aus dem Extras Menü von ZOC) kontaktiert, um Private-Keys für Login oder innere Sitzungen (wie bei OpenSSH Parameter `-A`) bereitzustellen.

GSSAPI Authentifizierung

Aktiviert oder deaktiviert die GSSAPI Authentifizierungs-Methode. Diese Methode wird üblicherweise für eine Authentifizierungs-Methode verwendet, welche den SSH-Server einen zentralen Host kontaktieren lässt, um zu prüfen, ob der Benutzer die Berechtigung zum Login hat.

GSSAPI Authentifizierungs-Methoden

Bei aktivierter GSSAPI Authentifizierungs-Methode kann zwischen zwei Implementierungen gewählt werden: Kerberos GSSAPI (unter Verwendung der GSSAPI-Bibliothek aus dem originalen MIT Kerberos Paket), oder Microsoft-SSPI (welche sich auf die Microsoft-Implementierung des Kerberos-Protokolls bezieht, was üblicherweise mit der Verwendung von Windows-Domänen und Active-Directory einhergeht).

Keyboard-Interactive Authentifizierung

Aktiviert oder deaktiviert die Keyboard-Interactive Authentifizierungs-Methode. Diese Methode wird üblicherweise für Challenge-Response-Authentifizierungen verwendet (z.B. mit SecureId Karten). Manchmal fragt diese Methode auch einfach nur nach dem Passwort.

Passwort Authentifizierung

Aktiviert oder deaktiviert die Passwort Authentifizierungs-Methode.

Tunnelprofile

Außerdem kann eine Tunnelprofildatei angegeben bzw. erstellt werden, die lokale und serverseitige Tunnels (Port Forwarding) enthält. Die dort eingestellten Tunnels werden dann beim Verbinden mit einem Host automatisch eingerichtet (siehe auch [Tunnelprofile erstellen](#)).

X11-Weiterleitung aktivieren

Erstellt einen Tunnel, der X11-Daten vom Server zum Client weiterleitet. Diese Option ist notwendig, wenn X11 Befehle über eine Remote-Shell ausgeführt werden sollen. Die Einstellung ist das Equivalent des OpenSSH-Parameter `-X`.

Remote-Befehl

Ein hier angegebener Befehl wird auf dem Remote-Rechner ausgeführt, statt in eine Shell zu verbinden. Diese Einstellung ist das Equivalent des Befehls 'command' von OpenSSH.

Weitere Optionen

Da die in ZOC verwendete SSH Implementierung auf OpenSSH basiert, können weitere Kommandozeilen-Parameter von OpenSSH hier angegeben werden, z.B. `-4`, um die Konnektivität auf IPv4 einzuschränken, oder `-C`, um die gzip-Kompression auf dem Datenkanal zu aktivieren. Bitte beachten Sie, dass nicht alle Optionen unterstützt werden, speziell wenn diese auf weiterführenden Funktionalitäten wie Proxy-Befehlen, Konfigurationsdateien, Multiplex-Mastern, etc. beruhen.

1.8.1.4 Globale SSH Authentifizierungsdateien

Falls Sie sich an unterschiedlichen Gegenstellen mit immer den gleichen Authentifizierungsdateien (Public/Private Key Dateien) anmelden, können diese im globalen SSH Options-Dialog eingegeben werden.

Diese Dateien werden zur Authentifizierung eingesetzt, wenn Sie eine Verbindung über "Neue Verbindung" oder über das Adressbuch aufbauen, und dort die Option "Globale SSH Schlüssel zur Authentifizierung verwenden" ausgewählt haben.

Alternativ können Sie per Option angeben, dass Sie globale SSH Schlüssel für jeder Verbindung verwenden möchten. In diesem Fall müssen Sie diese Einstellung nicht mehr für jede einzelne Verbindung vornehmen.

Den globalen Authentifizierungs-Dialog finden Sie im Menü [Extras](#).

Siehe auch: [SSH→Optionen](#)

1.8.1.5 *Secure Shell (SSH) Menüfunktionen*

Der Verbindungstyp SSH fügt drei zusätzliche Funktionen ins ZOC-Menü [Datei](#) ein:

- ☐ [Erzeugen von Public/Private-Keys](#)
- ☐ [Ändern von Passphrases](#)
- ☐ [Tunnelprofile erstellen](#)

1.8.1.6 *REXX-Befehl ZocDeviceControl für SSH*

`ZocDeviceControl` ist ein [ZOC-REXX Befehl](#), der es erlaubt, Funktionen, die spezifisch für eine bestimmte Verbindungsmethode sind, auszuführen.

Verfügbare Unterbefehle für den `ZocDeviceControl` Skriptbefehl in Bezug auf SSH sind:

RESOLV <hostname oder ip>

Liefert einen String in der Form "<hostname>; <ip>" mit den Daten des angefragten Hosts, z.B. `SAY ZocDeviceControl("RESOLV mail.emtec.com")`

1.8.1.7 *Erzeugen von Public/Private-Keys*

Hinweis: Dieses Fenster bietet eine bequeme Möglichkeit, die am häufigsten verwendeten öffentlichen/privaten Schlüssel zu erstellen (einschließlich nicht-residenter FIDO2-Schlüssel). Wenn Sie eine feinere Kontrolle über die Schlüsselerstellung benötigen, finden Sie im ZOC-Ausführungsordner ein Befehlszeilenprogramm, das das genaue Verhalten des OpenSSH-Befehls `ssh-keygen` nachahmt (inclusive Support für die `ecdsa-sk` und `ed25519-sk` keys). Dieses Programm finden Sie unter `C:\Program Files\ZOC9ssh-keygen.exe`

Im Fenster 'SSH Authentifizierungsdateien erzeugen' können SSH2-Authentifizierungsdateien (Public/Private Keys (Schlüssel)) für die spätere Verwendung in Adressbucheinträgen oder der manuellen Anwahl erzeugt werden.

Sie geben hierbei Typ und Länge des Schlüssels an (z.B. RSA 2048 Bit), so wie eine Passphrase mit der die Datei geschützt ist. Der Begriff Passphrase steht im Prinzip für ein Kennwort das verhindert, dass jemand, der in den Besitz des privaten Teils Ihres Schlüssels gelangt, diesen verwenden kann. Wegen der hohen Anforderung an Sicherheit wird hierbei jedoch meist nicht nur ein Wort, sondern ein kurzer Satz verwendet, deshalb der Begriff **Passphrase**

Der Schlüssel besteht aus zwei Teilen, einem privaten und einem öffentlichen, und wird in zwei Dateien gespeichert. Der private Teil ist durch die Passphrase geschützt und **darf keinesfalls weitergegeben werden**, muss aber im SSH-Verzeichnis von ZOC gespeichert werden.

Der öffentliche Teil hat die Endung `.pub` und wird auf den Server überspielt.

Dort wird er üblicherweise (das kann nach Art des Servers und der SSH Serversoftware variieren) im Ordner `~/.ssh` (ein versteckter Ordner in Ihrem Home-Verzeichnis am Server) abgelegt und dann an die Datei `authorized_keys` angehängt.

Mit ZOC kann dies auf einem Unix/Linux-Host wie folgt durchgeführt werden:

1. Kopieren Sie den Inhalt der neu erstellten öffentlichen Schlüsseldatei, z.B. `id_rsa.pub`, in die Zwischenablage, indem Sie die Funktion [Extras-Menü→SSH Public-Key in Zwischenablage kopieren](#) verwenden.

2. Melden Sie sich am Remote-Computer an und geben Sie die folgenden Befehle ein, wobei Sie nach dem Tippen von `echo` den Inhalt der Zwischenablage in den Befehl einfügen, indem Sie die Funktion [Bearbeiten-Menü→Einfügen](#) verwenden:

```
cd ~/.ssh
echo (Zwischenablage hier einfügen) >>authorized_keys
```

Siehe auch: [SSH→Optionen](#) und [SSH→Globale Optionen](#).

1.8.1.8 *Ändern von Passphrases*

Über diese Funktion ist es möglich, die Passphrase (also der kurze Satz, der als Kennwort für den privaten Teil des Schlüssels verwendet wird) für bestehende Authentifizierungsdateien zu ändern.

Siehe auch: [SSH](#) und [Die Key-Erzeugungsfunktion](#)

1.8.1.9 *Tunnelprofile erstellen*

Hier können Tunnelprofildateien erzeugt und geändert werden, die dann in den SSH-Optionen angegeben werden können.

Ein Tunnel ist eine TCP Verbindung zwischen Ihrem und dem entfernten Rechner, die verschlüsselt durch den SSH Kanal läuft. Dazu gehört auf einer Seite eine Portnummer zu der hin Verbindungen aufgebaut werden und auf der anderen Seite eine Rechneradresse und Port, zu der die Verbindung weitergeleitet wird.

Ein Tunnelprofil kann eine oder mehrere Tunnel für eine SSH Verbindung beinhalten. Mit diesem Dialog können Sie Tunnelprofildateien laden, die darin enthaltenen Tunnel bearbeiten (neue hinzufügen, vorhandene bearbeiten oder löschen) und die Profildatei abspeichern. Der Name der Profildatei wird später bei einer SSH Verbindung in den Optionen angegeben, damit bei Aufbau der Verbindung die im Profil enthaltenen Tunnel erstellt werden.

Um z.B. eine gesicherten Email Abruf über den SSH Kanal zu machen, könnten Sie einen lokalen Tunnel einrichten. Der lokale Port könnte 10110 sein, das entfernte Ziel wäre dann der Name des Mailservers und der Port, z.B. `mail.irgendwo.de:110`. Wenn dies eingerichtet ist, könnten Sie Ihrem Mailprogramm als Server den eigenen Rechner mit Portnummer 10110 angeben, also `127.0.0.1:10110` oder `localhost:10110`. Beim Mailabruf baut dann das Mailprogramm die Verbindung zu Ihrem Rechner auf und das laufende ZOC leitet die Daten über den SSH Kanal dann zum Mailserver weiter.

Siehe auch: [SSH→Optionen](#) und [Tunnelprofil-Einträge](#)

1.8.1.10 *Eintrag für Tunnelprofile*

Wie bei [Tunnelprofile erstellen](#) beschrieben, besteht ein Tunnel in der Regel aus einer Kombination von einem Port und einer Rechneradresse mit Port.

Neben solchen direkten Tunneln, unterstützt ZOC auch dynamische Tunnel (ZOC als SOCKS4 bzw. SOCKS5 Server) bei denen erst bei Verbindungsaufbau eines Clients zum Tunnel festgelegt wird, wohin die Verbindung auf der Serverseite aufgebaut wird.

Dieser Dialog legt einen Tunnelleintrag fest. Um das o.g. Beispiel einzurücken, wählen Sie "Tunnel von lokaler zu entfernter Adresse", geben als lokalen Port 10110 ein und füllen das Feld für das Ziel mit `mail.irgendwo.de:110`

Hinweis: Wie bei OpenSSH müssen IPv6 Adressen in eckigen Klammern angegeben werden, z.B. `[fc00::100:1]:10022`

1.8.1.11 *Auswahl eines SSH Schlüssels*

Für die Auswahl eines SSH-Schlüssel können Sie zwischen mehreren Optionen wählen.

OpenSSH Schlüssel/Key

Hier können OpenSSH-kompatible Private-Key Dateien oder Zertifikate angegeben werden. Falls Ihre Schlüssel im Putty ppk Format vorliegen, müssen Sie die Dateien mittels putty-keygen ins OpenSSH Format konvertieren.

PKCS#11 DLL/Bibliothek

PKCS#11 ist eine Schnittstelle, die hauptsächlich von Kartenlesern oder Hardware-Schlüsseln verwendet wird. Sie können eine Windows DLL oder macOS dylib/so Datei angeben, die ZOC dann erlaubt, auf den Hardware-Schlüssel zuzugreifen. Für die relativ populären Yubi-Keys, wäre das unter Windows z.B. die Datei `C:\Program Files\Yubico\Yubico PIV`

Tool\bin\libykeys11.dll.

Globale SSH Schlüsseldateien

ZOC bietet die Möglichkeit, [Globale Schlüsseldateien](#) zu definieren, entweder über das Extras-Menü, oder über das SSH-Einstellungs-Fenster. Wenn Sie diese Einstellung verwenden, wird ZOC alle dort angegebenen Schlüsseldateien verwenden, und sie dem Server zur möglichen Authentifizierung anbieten.

Windows-Zertifikat oder CAC/PIV Leser

Wenn Sie Schlüssel zur Authentifizierung im Windows-Zertifikatsspeicher abgelegt haben, können Sie hier einen dieser Schlüssel auswählen. Dies ermöglicht es auch, Hardware wie z.B. CAC/PIV-Leser oder USB-Hardwareschlüssel zu verwenden, welche damit verwendbare Schlüssel im Windows-Zertifikatsspeicher hinterlegen.

1.8.2 Telnet

Die Verbindungsmethode Telnet erlaubt den Zugriff auf Gegenstellen über das Internet oder eine LAN-Verbindung anhand des sogenannten Telnet-Protokolles.

Telnet überträgt unverschlüsselte Daten und wurde inzwischen fast vollständig durch das [Secure Shell](#) Protokoll ersetzt.

Die folgenden Themen behandeln die Verwendung von Telnet-Verbindungen in ZOC:

- ☐ [Telnet Überblick](#)
- ☐ [Telnet Einstellungen](#)
- ☐ [Menüfunktionen](#)
- ☐ [REXX ZocDeviceControl](#)

1.8.2.1 Telnet Überblick

Der Verbindungstyp Telnet bietet Telnet-Zugriff auf Hosts über eine bestehende Internetverbindung.

Um Verbindung zu einem Host herzustellen, wird in einem Adressbucheintrag oder unter Manuelle Wahl der Hostname oder die Internet-Adresse anstelle einer Telefonnummer angegeben. ZOC versucht dann, über die laufende Internetverbindung einen Kontakt zur Gegenstelle herzustellen, z.B.

`bbs.channell1.com`

Verbindungen zum entfernten Rechner werden über den Standard Telnet-Port (23) hergestellt. Soll über einen anderen Port verbunden werden, kann die Portnummer oder der Servicenamen nach dem Hostnamen oder der Adresse, mit Punkt oder Leerzeichen getrennt, angegeben werden, z.B.

`mail.hogwarts.edu 25` oder `mail.hogwarts.edu:25` oder `mail.hogwarts.edu smtp`

Falls Sie eine NO RESPONSE Fehlermeldung bekommen, können Sie versuchen, einen * (Stern) vor den Hostnamen (oder die Adresse) zu schreiben. In diesem Fall versucht ZOC nicht den Host zu PINGen, bevor die Verbindung aufgebaut wird (dies kann helfen, wenn der Host sehr lange braucht um sich zu melden): `*slowhost.leisure.net`

Falls die Gegenstelle kein echter Telnet Server ist (d.h. ein Rechner der eines der normalen TCP Protokolle wie SMTP, POP3, etc. verwendet), können Sie in den [Telnet Einstellungen](#) [Pure Socket Verbindung](#) aktivieren.

1.8.2.2 Telnet Einstellungen

Um Optionen für Telnet-Verbindungen anzupassen, navigieren Sie zum Abschnitt Verbindungstyp im [Verbindungsprofil](#).

Diese Änderungen gelten dann für alle Verbindungen, die dieses Verbindungsprofil verwenden. Wenn Sie

die Optionen für eine einzelne Verbindung ändern möchten, klicken Sie auf die Schaltfläche [Konfigurieren](#) im [Neue Verbindung](#)-Fenster.

Für Verbindungen, die über das [Adressbuch](#) hergestellt werden, verwenden Sie die Schaltfläche [Konfigurieren](#) im [Host-Tab](#) Ihres Adressbuch-Eintrags.

Nachfolgende Optionen sind für Telnet Verbindungen verfügbar:

Socks Proxy

Wenn Sie zum Erreichen Ihrer Telnet-Gegenstelle einen Proxy verwenden müssen, können Sie hier den Namen oder die IP-Adresse und optional einen Port angeben (Standard ist Port 1080). Wenn der Proxy ein SOCKS5 Proxy ist, können Sie Benutzernamen und Passwort im Format `user:pass@ip:port` angeben.

Reine IP-Socket Verbindung, kein Telnet

Diese Option baut eine reine TCP Verbindung zur Gegenstelle auf, d.h. es werden keine spezifischen Telnet-Optionen ausgehandelt. Dies ist sinnvoll, wenn die Gegenstelle kein Telnet-Server ist (z.B. SMTP oder andere RFC Server).

TCP 'Keep Alive' Option aktivieren

Wenn diese Option aktiviert ist, konfiguriert das TCP Keep-Alive feature für die TCP Verbindung mit dem Host (technisch gesehen wird dafür die SO_KEEPALIVE Option auf dem Socket aktiviert). Zusätzlich wird dann alle 60 Sekunden ein TELNET NOP Paket (FFh F1h) geschickt.

Falls das nicht funktioniert, können Sie auch die Timeout Einstellung im Verbindungsprofil (Abschnitt Terminal) verwenden (siehe [Tutorial: Configure Timeout Settings](#) (engl.)).

Mit lokalem Echo starten

Bei eingeschalteter Funktion verwendet ZOC beim Beginn der Verbindung ein lokales Echo (anstatt dem langsameren Gegenstellenecho). Allerdings wird diese Option beim Login-Prozess oftmals von der Gegenstelle überschrieben (d.h. zurück auf Remote-Echo gestellt).

Host darf lokales Echo des Terminal ändern

Erlaubt der Gegenstelle, die Einstellung für lokales Echo zu ändern (Verbindungsprofil→Terminal).

Verbindung bei Fehler halten

Normalerweise legt ZOC bei einem Telnet-Übertragungsfehler auf. Wenn diese Option eingeschaltet ist, bleibt die Verbindung auch bei Übertragungsfehlern bestehen.

ERWEITERTE OPTIONEN

Keine Hostnamen zu IP Adressen suchen

Beim Verbindungsaufbau per IP Adresse sucht ZOC normalerweise den dazu passenden Hostnamen. Dies kann abgeschaltet werden, falls es nicht erwünscht ist (z.B. weil der DNS Server gebührenpflichtig ist).

Nur IP-V4 Verbindung

Diese Einstellung legt fest, dass der DNS-Lookup nur IP-V4 Adressen verwendet.

CR-Umsetzung beim Senden

Diese Option muss eingeschaltet werden, wenn die Gegenstelle ein Nullzeichen nach einem CR-Zeichen, dem kein LF-Zeichen folgt, benötigt.

Host sendet CR/NUL im Binärmodus

Diese Option kann verwendet werden, wenn Binärtransfers (wie z.B. Zmodem) über die Telnet-Verbindung nicht funktionieren.

Telnet-Option 'Do SGA' anbieten

SGA ist eine Funktion des Telnet-Protokolls. Normalerweise wird diese von ZOC deaktiviert, was mit 99.9% aller Gegenstellen keinen Unterschied macht. Falls Sie beim Anmelden jedoch eine Fehlermeldung im Bezug auf SGA erhalten, versuchen Sie bitte diese Option zu aktivieren.

Im Binärmodus starten

Mit dieser Option versucht ZOC mit dem Host von Anfang an eine Binärverbindung zu vereinbaren.

Port/Adresse für Eingehende Verbindungen

Hier kann angegeben werden, auf welchem Port ZOC Telnet-Verbindungen annimmt, wenn die Verbindungsannahme aktiviert ist (Funktion [Verbindungen annehmen](#) im Dateimenü).

Terminal Geschwindigkeit

Hier kann die Geschwindigkeit des Terminals für beide Richtungen in Baud angegeben werden (z.B. 9600,9600).

1.8.2.3 Telnet Menüfunktionen

Der Verbindungstyp Telnet fügt drei Funktionen in das [Datei](#) Menü von ZOC ein.

Are you there?

Diese Funktion versucht, eine Antwort von der Gegenstelle zu erzwingen (normalerweise [\[YES\]](#)), um herauszufinden, ob noch eine Verbindung zum Host besteht.

Interrupt Process

Diese Funktion bricht ein Programm ab, das gerade am anderen Rechner ausgeführt wird. Diese Funktion entspricht dem lokalen Drücken der Tasten Strg+Break.

Abort Output

Diese Funktion bricht eine Ausgabe der Gegenstelle ab (z.B. beim Anzeigen einer sehr langen Textdatei).

1.8.2.4 REXX ZocDeviceControl für Telnet

`ZocDeviceControl` ist ein [ZOC-REXX Befehl](#), der es erlaubt, Funktionen, die spezifisch für eine bestimmte Verbindungsmethode sind, auszuführen.

Verfügbare Unterbefehle für den `ZocDeviceControl` Skriptbefehl in Bezug auf Telnet sind:

RESOLV <hostname oder ip>

Liefert einen String in der Form "<hostname>; <ip>" mit den Daten des angefragten Hosts, z.B. `SAY ZocDeviceControl("RESOLV mail.emtec.com")`

TCP_NODELAY ON

Aktiviert die TCP_NODELAY Option auf dem unterliegenden Socket (googlen Sie nach TCP_NODELAY und/oder Nagle's Algorithmus), z.B. `CALL ZocDeviceControl "TCP_NODELAY ON"`

1.8.3 Telnet/SSL

Telnet/SSL ist eine Verbindungsmethode, die eine verschlüsselte Telnet-Verbindung über unsichere Netzwerke ermöglicht. Das Protokoll basiert auf [Telnet](#), das mit einer SSL-Schicht verschlüsselt wird. Die folgenden Themen behandeln die Verwendung von Telnet/SSL-Verbindungen in ZOC:

- ❑ [Telnet/SSL Überblick](#)
- ❑ [Telnet/SSL Einstellungen](#)
- ❑ [REXX ZocDeviceControl](#)

1.8.3.1 *Telnet/SSL Überblick*

Die Verbindungsmethode Telnet/SSL ist prinzipiell mit [Telnet](#) identisch. Einziger Unterschied ist die Tatsache, dass der Datenverkehr durch eine SSL-Schicht verschlüsselt wird.

Mit Hilfe von Telnet/SSL können Sie mit SSL Gegenstellen (wie z.B. mit TLS/SSL gesicherten IBM Mainframes oder mit Webservern auf Port 443) kommunizieren.

1.8.3.2 *Telnet/SSL Optionen*

Um Optionen für Telnet-Verbindungen anzupassen, navigieren Sie zum Abschnitt Verbindungstyp im [Verbindungsprofil](#).

Diese Änderungen gelten dann für alle Verbindungen, die dieses Verbindungsprofil verwenden. Wenn Sie die Optionen für eine einzelne Verbindung ändern möchten, klicken Sie auf die Schaltfläche [Konfigurieren](#) im [Neue Verbindung](#)-Fenster.

Für Verbindungen, die über das [Adressbuch](#) hergestellt werden, verwenden Sie die Schaltfläche [Konfigurieren](#) im [Host-Tab](#) Ihres Adressbuch-Eintrags.

Die meisten Optionen sind mit denen von Telnet (siehe [Telnet Optionen](#)) identisch.

Der Verbindungstyp Telnet/SSL ist prinzipiell mit [Telnet](#) identisch. Einziger Unterschied ist die Tatsache, dass der Datenverkehr durch eine SSL-Schicht verschlüsselt wird.

Telnet/SSL bietet die Möglichkeit der Authentifizierung am Server mittels eines Client-Zertifikates. Sollte die Client-Zertifikats-Datei keinen privaten Schlüssel enthalten, kann dieser auch über eine separate Private-Key Schlüsseldatei angegeben werden.

Weiterhin kann das Sicherheits-Protokoll auf TLS 1.2 beschränkt werden. Andernfalls akzeptiert ZOC auch SSLv2, SSLv3, TLS 1.0 und TLS 1.1 als mögliche Sicherheits-Protokolle.

1.8.3.3 *Telnet/SSL ZocDeviceControl für Telnet/SSL*

`ZocDeviceControl` ist ein [ZOC-REXX Befehl](#), der es erlaubt, Funktionen, die spezifisch für eine bestimmte Verbindungsmethode sind, auszuführen.

Verfügbare Unterbefehle für den `ZocDeviceControl` Skriptbefehl in Bezug auf Telnet/SSL sind identisch mit denen für [Telnet](#).

1.8.4 Serial/Direct und Serial/Modem

Serial/Direct und Serial/Modem sind Verbindungsmethoden in ZOC, die lokale serielle Verbindungen (Serial/Direct) bzw. ferne serielle Verbindungen (Serial/Modem) verwenden, um mit Geräten oder Gegenstellen zu kommunizieren.

Die folgenden Themen behandeln die Verwendung von seriellen Verbindungen in ZOC:

- ❑ [Serial/Modem und Serial/Direct Überblick](#)
- ❑ [Serial/Modem und Serial/Direct Einstellungen](#)
- ❑ [REXX ZocDeviceControl](#)
- ❑ [AT Befehle Bearbeiten](#)

1.8.4.1 **Serial/Modem Überblick**

Dieser Abschnitt betrifft zwei sehr ähnliche Kommunikationsmethoden: [Serial/Modem](#) und [Serial/Direct](#). Serial/Modem erlaubt den Aufbau Verbindungen über Modems mit AT-Befehlssatz, während Serial/Direct für den Zugriff auf andere serielle Geräte dient, die direkt an den seriellen Anschluss des Rechners (z.B. Windows `COM2` oder macOS `/dev/cu.pl2303serial`) angeschlossen sind.

Falls Sie vorkonfigurierte Modems (oder modem-ähnliche Geräte, z.B. Mobiltelefone) an Ihrem Rechner angeschlossen haben, ist es unter Umständen einfacher diese in ZOC als [Windows-Modems](#) anzusprechen, um damit Verbindungen aufzubauen, speziell wenn diese verwendet werden sollen, um eine Wählverbindung zu einem anderen Rechner aufzubauen, anstatt direkt mit dem Gerät selbst zu kommunizieren (z.B. um es zu konfigurieren).

Siehe auch die Hinweise zum Verbindungsaufbau bei [Neue Verbindung](#)

1.8.4.2 **Serial/Modem Einstellungen**

Um Optionen für serielle Verbindungen anzupassen, navigieren Sie zum Abschnitt Verbindungstyp im [Verbindungsprofil](#).

Diese Änderungen gelten dann für alle Verbindungen, die dieses Verbindungsprofil verwenden. Wenn Sie die Optionen für eine einzelne Verbindung ändern möchten, klicken Sie auf die Schaltfläche [Konfigurieren](#) im [Neue Verbindung](#)-Fenster.

Für Verbindungen, die über das [Adressbuch](#) hergestellt werden, verwenden Sie die Schaltfläche [Konfigurieren](#) im [Host-Tab](#) Ihres Adressbuch-Eintrags.

Die nachfolgende Liste beschreibt alle Optionen der Verbindungstypen Serial/Modem und Serial/Direct.

Com-Port

Hier wird der Name des Com-Ports eingegeben, z.B. unter Windows `COM1` (bitte kein Leerzeichen zwischen COM und 1). oder unter macOS `/dev/cu.pl2303serial`

Windows: Zum Ansprechen von COM-Ports höher als 8 muss ab Windows NT die Schreibweise `\\.\COMxx` verwendet werden. Zur Verwendung eines Servermodems unter Windows ist ein Modem-Sharing Programm notwendig, (z.B. Stomper).

Übertragung

Hier kann eingestellt werden, in welcher Weise Daten zwischen Computer und Modem ausgetauscht werden (nicht zu verwechseln mit der Weise, wie Daten zwischen den Modems über die Telefonleitung geschickt werden). Es muss die Übertragungsgeschwindigkeit (Bits pro Sekunde), die Anzahl der Datenbits, die Parität (N, E, O, M, S) und die Anzahl der Stopbits eingestellt werden.

Diese Einstellungen hängen stark von der Gegenstelle ab (und natürlich von den Möglichkeiten des Modems).

Oftmals werden diese Einstellungen in abgekürzter Version angegeben (z.B. in Beschreibungen von Mailboxen). 38400-8N1 bedeutet also 38400 Bits pro Sekunde, 8 Datenbits, keine Parität und ein Stopbit.

RTS/CTS Steuerung

Wenn aktiv, erfolgt die Modemsteuerung über 7 Leitungen.

RTS/CTS wird verwendet, um den Datenfluss zwischen Computer und Modem zu steuern und ermöglicht beiden Seiten, sich gegenseitig beim Senden zu unterbrechen.

Diese Funktion ist unerlässlich für schnelle Übertragungen, wenn die Geschwindigkeit zwischen

Computer und Modem sich von der Geschwindigkeit zwischen Modem und Modem unterscheidet (wie z.B. bei MNP5 und V.42 Modems). Auch in einer Multitaskingumgebung ist RTS/CTS sinnvoll, wenn der Prozessor bei gerade ankommenden Daten schon stark belastet ist.

Deshalb ist es empfehlenswert, RTS/CTS einzuschalten. Aber auch das Modem muss diese RTS/CTS Steuerung unterstützen, deshalb muss bei der Modem-Konfiguration darauf geachtet werden, dass RTS/CTS aktiviert ist. Das entsprechende Modem-Kommando kann auch beim Init-Kommando des Modems mitüberegeben werden (siehe [Einstellungen Modem](#)).

DSR Steuerung

Die DSR/DTR Steuerung ist ähnlich der RTS/CTS Steuerung eine Hardware-Steuerung. Allerdings wird hier nicht der Datenfluss kontrolliert, sondern die Verfügbarkeit von Computer und Modem überprüft (z.B. ob das Modem gerade ein- oder ausgeschaltet ist).

Wenn aktiv, überprüft ZOC das DSR-Signal vom Modem. Diese Option sollte nur aktiv sein, wenn Modem und Verbindungskabel DSR/DTR unterstützen. Ansonsten findet keine Übertragung zwischen Rechner und Modem statt und ZOC wird blockiert.

XOn/XOff Steuerung

Wenn aktiv, verwendet ZOC das XON/XOFF Steuerungs-Protokoll (eine weitere Methode, um den Datenstrom zu kontrollieren). Hierbei werden spezielle Zeichen verwendet, um den Sender anzuhalten. Diese Methode ist schlechter als die Steuerung mit dem RTS/CTS Protokoll und sollte deshalb nur in Sonderfällen eingeschaltet werden.

CD Prüfung

Diese Option sollte aktiv sein, wenn das Modem das Carrier Detect (CD) Signal unterstützt (bei den meisten Modems geschieht dies mit dem Kommando `AT+C1`).

ZOC benötigt dieses Signal für die Steuerung des Timers, zur Berechnung der Gebühren sowie zum Schreiben des Anrufprotokolls, das in [Optionen→Programmeinstellungen→Spez. Dateien](#) angegeben ist.

Dauer Break-Signal

Hier wird die Länge eines Break-Signals angegeben (einige Gegenstellen verwenden Breaks, um eine Verarbeitung abubrechen). Ein Break wird mit Strg+Ende gesendet. Die Dauer eines Breaks wird in Millisekunden angegeben und liegt üblicherweise zwischen 250 und 400 ms.

AT-Befehle

Um Modems anzusteuern werden spezielle Kommandos verwendet (sog. AT-Befehle). Sie können im Serial/Modem Optionsdialog ein Profil mit verschiedenen AT-Kommandos anlegen und auswählen bzw. unterschiedliche Profile verwenden um ggf. mit unterschiedlichen Modems zu arbeiten. Siehe [Modem Einstellungen](#) PT(COM-Port freigeben wenn das Fenster minimiert ist). Zusätzlich können Sie eine Option festlegen, um den seriellen Port freizugeben, wenn das ZOC-Fenster minimiert ist. Dies würde es Ihnen ermöglichen, den COM-Port von einer anderen Software aus zu nutzen, während das ZOC-Fenster minimiert ist. Bitte beachten Sie, dass die Option "CD-Signal ist gültig" aktiviert sein muss und der Port nur freigegeben wird, wenn keine aktive Modemverbindung besteht.

1.8.4.3 REXX ZocDeviceControl für serielle Verbindungen

`ZocDeviceControl` ist ein [ZOC-REXX Befehl](#), der es erlaubt, Funktionen, die spezifisch für eine bestimmte Verbindungsmethode sind, auszuführen.

Verfügbare Unterbefehle für den `ZocDeviceControl` Skriptbefehl in Bezug auf serielle Verbindungen sind:

GETRS232SIGNALS

Liefert einen String, der die gesetzten eingehenden Signale/Pins (CTS, DSR, CD, RI) am seriellen

Port indiziert, z.B. `##[CTS][DSR][CD]##`

Hinweis: Zur Kompatibilität mit früheren Versionen von ZOC, kann auch das Kommando GETSTATE verwendet werden.

SETRTS ON|OFF

Setzt den Status des RTS Signals am RS23 Port (funktioniert nur, wenn in den seriellen Einstellungen Rts/Cts nicht auf Handshake gesetzt ist), z.B. `Call ZocDeviceControl("SETRTS ON")`

SETDTR ON|OFF

Setzt den Status des DTR Signals am RS23 Port (funktioniert nur, wenn in den seriellen Einstellungen DTR/DSR nicht auf Handshake gesetzt ist), z.B. `Call ZocDeviceControl("SETDTR OFF")`

TESTACCESS <name>

Dieser Befehl testet ob der Port <name> (z.B. `SAY ZocDeviceControl("TESTACCESS COM3")` oder `SAY ZocDeviceControl("TESTACCESS /dev/cu.pl2303")`) gültig ist und ob er zur Kommunikation benutzt werden könnte.

FINDUSBPORT <text>

Dieser Befehl sucht den ersten USB-Port, bei dem <text> in der Device-Beschreibung auftaucht (z.B. `port= ZocDeviceControl("FINDUSBPORT FTDI")`).

1.8.4.4 Modem Einstellungen

Auf dieser Seite können die von ZOC benötigten Modem-Kommandos angegeben werden. Diese Befehle können Steuerzeichen (z.B. `^M`, siehe [Steuerzeichen](#)), das `~` Zeichen (1/3 sek. Verzögerung) und `^#` als Platzhalter für die Telefonnummer enthalten.

Initialisierung

Hier können mehrere Init-Kommandos angegeben werden, die das Modem vor Anwahl einer Gegenstelle ausführt. Es muss beachtet werden, dass ZOC (im Gegensatz zu einigen anderen Programmen) ein `^M` am Ende eines Init-Kommandos benötigt.

Abhängig vom System kann es sinnvoll sein, bei jedem Laden eines Verbindungsprofils das Modem neu zu initialisieren (so kann das Modem für jedes Verbindungsprofil entsprechend initialisiert werden).

Normalerweise sendet ZOC kein Init-Kommando, wenn ein Carrier Detect Signal vom Modem vorliegt (um einen Modem-Reset während einer Verbindung zu vermeiden). Einige Modems senden allerdings immer ein CD Signal. In diesem Fall sollte die Option 'Auch bei Carrier Detect initialisieren' eingeschaltet sein.

Wählkommandos

In ZOC können vier verschiedene Wählkommandos angegeben werden, um Telefonnummern anzuwählen (z.B. mit und ohne Nebenstellen-0 am Anfang). Welches der vier Wählkommandos verwendet werden soll, kann pro Eintrag im [Adressbuch](#) und beim manuellen Wählen im [Datei-Menü](#) angegeben werden. Um eine Telefonnummer mit dem Modem zu wählen, muss ein Terminalprogramm den Wählbefehl, die Telefonnummer und ein `^M` an das Modem schicken. Der Wählbefehl lautet für Tonwahl `ATDT`, für Pulswahl `ATDP` und für Zyxel ISDN-Modems `ATDI`.

ZOC verwendet beim Wählkommando den Platzhalter `^#`, der durch die zu wählende Nummer ersetzt wird. Bei `ATDT ^#^M` wird also erst ATDT, dann die zu wählende Nummer und dann Return gesendet. Durch Verwendung des Modembefehls `X3` wird erreicht, dass das Modem auch dann mit dem Wählen beginnt, wenn nach dem Abheben noch kein Freizeichen erkannt wird (wie das z.B. bei

Telefonanlagen der Fall ist). In diesem Fall lautet der Wählenstring `ATX3DT 0,^#^M`. Damit wird erst die Freizeichen-Erkennung ausgeschaltet, dann eine Null vorgewählt und dann die zu wählende Nummer mit abschließendem Return gesendet.

Soll vor dem Wählenkommando noch ein weiterer Befehl ausgeführt werden, sollten zwischen den Befehlen mehrere ~-Zeichen angegeben werden, also z.B. `ATZ^M~~~ATDT ^#^M`, um dem Modem Zeit für die Abarbeitung des ersten Befehls zu geben.

Auflegen

ZOC bietet zwei Methoden an, um die Verbindung zu unterbrechen. Die Verwendung des DTR-Signals ist die elegantere von beiden. Ansonsten verwendet ZOC das ATH Kommando.

Auto Answer

Hier werden zwei Modem-Befehle verwendet, um den Auto Answer Modus des Modems ein- oder auszuschalten. Diese Befehle `ATS0=1^M` und `ATS0=0^M` sind für alle Modems gleich.

1.8.5 Windows Modems

Windows Modems ist ein Kommunikations-Mechanismus, der innerhalb von Microsoft Windows registrierte Modems verwendet, um mit entfernten Computern zu kommunizieren.

Die folgenden Themen behandeln die Verwendung von Windows Modems in ZOC:

- ☐ [Überblick](#)
- ☐ [Windows Modems Einstellungen](#)
- ☐ [REXX ZocDevControl](#)

1.8.5.1 Windows-Modem Überblick

Mit diesem Verbindungstyp ist der Zugriff auf Modems möglich, die von Windows direkt per Betriebssystem installiert sind (sogenannte TAPI-Geräte). Die Optionen dieses Verbindungstyps zeigen eine Liste aller im System erkannten Windows-Modems an. Aus dieser Liste kann dann ein Modem ausgewählt und anschließend verwendet werden. Informationen zu ISDN-Karten, die als Windows Modems ins System eingebunden wurden, aber nicht funktionieren, finden Sie unter [Häufige Probleme, Fragen und Antworten](#).

1.8.5.2 Windows-Modem Einstellungen

Um Optionen für Modem-Verbindungen anzupassen, navigieren Sie zum Abschnitt Verbindungstyp im [Verbindungsprofil](#).

Diese Änderungen gelten dann für alle Verbindungen, die dieses Verbindungsprofil verwenden. Wenn Sie die Optionen für eine einzelne Verbindung ändern möchten, klicken Sie auf die Schaltfläche [Konfigurieren](#) im [Neue Verbindung](#)-Fenster.

Für Verbindungen, die über das [Adressbuch](#) hergestellt werden, verwenden Sie die Schaltfläche [Konfigurieren](#) im [Host-Tab](#) Ihres Adressbuch-Eintrags.

1.8.5.3 REXX ZocDeviceControl für Windows-Modems

`ZocDeviceControl` ist ein [ZOC-REXX Befehl](#), der es erlaubt, Funktionen, die spezifisch für eine bestimmte Verbindungsmethode sind, auszuführen.

Verfügbare Unterbefehle für den `ZocDeviceControl` Skriptbefehl in Bezug auf Windows Modems sind:

GETMODEMID <modemname>

Gibt eine ID zurück, die für ZocSetDevParm verwendet werden kann, um das Modem anzuwählen.

Beispiel:

```
modem= "Microlink 33.6TS PnP"  
id = ZocDeviceControl("GETMODEMID "||modem)  
Call ZocSetDevParm "[7]"id"|0|1:"
```

1.8.6 Local Shell

Innerhalb von ZOC ist Local Shell eigentlich keine Verbindungsmethode, sondern erlaubt es, ein lokales Terminal innerhalb eines ZOC Registerblatts zu öffnen.

Die folgenden Themen behandeln die Verwendung von Local Shell in ZOC:

- ☐ [Local Shell Überblick](#)
- ☐ [Local Shell Einstellungen](#)
- ☐ [REXX ZocDevControl](#)

1.8.6.1 Local Shell Überblick

Der Verbindungstyp Local Shell baut eine Pseudo-Verbindung zur Shell des lokalen Rechners aus. Auf dieser Weise kann über ZOC auf eine lokale Befehlszeile (z.B. bash oder zsh) zugegriffen werden, z.B. als Alternative zum macOS Terminal oder Windows DOS-Fenster.

Da die Kommunikation unter ZOC immer als Verbindung interpretiert wird, muss auch hier ein Verbindungsziel angegeben werden, auch wenn tatsächlich keine Verbindung zu einem anderen Computer aufgebaut wird. Deshalb muß bei Local Shell Sessions als im Feld [Verbinden mit](#) entweder `localhost` oder `127.0.0.1` angegeben werden.

Somit führt die Local Shell dann am lokalen Rechner den Befehl aus, der unter [Local Shell Einstellungen](#) eingetragen wurde (die Voreinstellung ist `/usr/bin/login` bzw. `C:\Windows\System32\cmd.exe`).

Um einen anderen Befehl auszuführen, können Sie diese Option im [Verbindungsprofil](#) im Abschnitt [Verbindungstyp](#) ändern. Für eine einzelne Verbindung können Sie dies auch mit dem [Konfigurieren](#) Knopf im Fenster [Neue Verbindung](#) tun, bzw. im Registerblatt [Gegenstelle](#) eines [Adressbuch](#)-Eintrags.

Alternativ können Sie im [Verbinden mit](#) Feld auch `#!` und den Namen eines beliebigen Systembefehls als Ziel für die Verbindung angeben. Beispiele unter macOS wären die Eingabe von `#!/usr/bin/login -p harry` oder `#!/usr/bin/zsh` oder `#!/usr/bin/ssh secure.hogwarts.edu` im das Feld [Verbinden mit](#). Unter Windows können Sie z.B. die Powershell in der Form `#!`

`C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe` verwenden.

1.8.6.2 Local Shell Einstellungen

Um Optionen für Local-Shell-Verbindungen anzupassen, navigieren Sie zum Abschnitt [Verbindungstyp](#) im [Verbindungsprofil](#).

Diese Änderungen gelten dann für alle Verbindungen, die dieses Verbindungsprofil verwenden. Wenn Sie die Optionen für eine einzelne Verbindung ändern möchten, klicken Sie auf die Schaltfläche [Konfigurieren](#) im [Neue Verbindung](#)-Fenster.

Für Verbindungen, die über das [Adressbuch](#) hergestellt werden, verwenden Sie die Schaltfläche [Konfigurieren](#) im [Host-Tab](#) Ihres Adressbuch-Eintrags.

Der Verbindungstyp Local Shell bietet folgende Einstellungsmöglichkeiten:

Local Shell

Wenn Local Shell mit der Zieladresse `localhost` oder `127.0.0.1` aufgerufen wird, führt ZOC letztlich einen Befehl aus und leitet den ein- und ausgehenden Datenstrom auf das ZOC-Fenster um.

Unter macOS wird standardmäßig von ZOC der Befehl `/usr/bin/login` ausgeführt, zusammen mit einem Parameter, der die Login-Abfrage umgeht und stattdessen den aktuell angemeldeten Benutzer zu verwenden: `/usr/bin/login -p -f $(USERNAME)`

Unter Windows wird `C:\Windows\System32\cmd.exe` aufgerufen. Um eine andere Shell verwenden zu können, muss diese Einstellung entsprechend geändert werden. Es kann stattdessen aber eine andere Shell, z.B. `#!C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe` angegeben werden.

1.8.6.3 REXX ZocDeviceControl für Local-Shell

`ZocDeviceControl` ist ein [ZOC-REXX Befehl](#), der es erlaubt, Funktionen, die spezifisch für eine bestimmte Verbindungsmethode sind, auszuführen.

Es gibt keine Unterbefehle für den `ZocDeviceControl` Skriptbefehl in Bezug auf Local Shell.

1.8.7 UDP (Datagramm)

UDP-Datagramm ist ein Mechanismus innerhalb lokaler und öffentlicher Netzwerke, bei dem Pakete zwischen zwei IP-Adressen ausgetauscht werden, ohne das Eintreffen der Übertragung zu verfolgen.

- ☐ [UDP/Datagramm Übersicht](#)
- ☐ [UDP/Datagramm Optionen](#)
- ☐ [REXX ZocDevControl](#)

1.8.7.1 UDP/Datagramm Übersicht

Die UDP/Datagramm-Methode ermöglicht es Ihnen, UDP-Netzwerkpakete mit einem anderen Computer auszutauschen. Während die meisten UDP-basierten Internetdienste wie DNS oder DHCP binäre Formate verwenden, die für die Verwendung mit ZOC nicht geeignet sind, gibt es andere Anwendungen, bei denen es notwendig sein kann, dass ZOC solche Pakete sendet oder empfängt, z.B. ein UDP-Logger oder ein UDP-Kurznachrichtendienst.

Um eine andere IP-Adresse als Gegenstelle für UDP festzulegen, verwenden Sie [Neue Verbindung](#) von ZOC oder erstellen Sie einen Adressbucheintrag und geben Sie die IP-Adresse des anderen Endpunkts im [Verbinden mit](#) Feld an, z.B. `192.168.1.10`.

Nachdem Sie auf [Verbinden](#) klicken, wartet ZOC auf UDP-Pakete von dieser Adresse und zeigt deren Inhalt auf dem Bildschirm. Die Portnummer, auf der ZOC auf diese Pakete hört, wird in den UDP-Einstellungen von ZOC konfiguriert.

Zeichen, die Sie eingeben, werden als einzelne UDP-Pakete an die Gegenstelle gesendet.

1.8.7.2 UDP/Datagramm Optionen

Wenn Sie eine Verbindung für den UDP-Verbindungstyp konfigurieren, können Sie die folgenden Optionen verwenden:

Mit lokalem Echo starten

Diese Option aktiviert die Option Lokales Echo im Sitzungsprofil von ZOC. Alle Zeichen, die Sie eingeben und die an die andere Adresse gesendet werden, werden auch auf dem Bildschirm angezeigt.

Port/Adresse für eingehende Verbindungen

Eine Portnummer oder eine Kombination aus lokaler IP-Adresse und Port, auf der ZOC auf eingehende Pakete hört, z.B. 10024 oder 192.168.1.100:10024

1.8.7.3 REXX ZocDeviceControl für UDP

`ZocDeviceControl` ist ein [ZOC-REXX Befehl](#), der es erlaubt, Funktionen, die spezifisch für eine bestimmte Verbindungsmethode sind, auszuführen.

Es gibt keine Unterbefehle für den `ZocDeviceControl` Skriptbefehl in Bezug auf UDP-Verbindungen.

1.8.8 Named-Pipe

Named-Pipe ist ein Kommunikations-Mechanismus innerhalb von Microsoft Windows, der es zwei lokalen Programmen (Prozessen) erlaubt, miteinander zu kommunizieren.

Die folgenden Themen behandeln die Verwendung von Named-Pipe in ZOC:

- ☐ [Named-Pipe Überblick](#)
- ☐ [Named-Pipe Einstellungen](#)
- ☐ [REXX ZocDevControl](#)

1.8.8.1 Named Pipe Überblick

Mit dem Verbindungstyp Named-Pipes kann auf Anwendungen zugegriffen und diese gesteuert werden, die eine zeichenorientierte Named-Pipe auf dem lokalen Rechner oder einem Netzwerk-Server unterstützen.

Um eine Verbindung zu einer Named-Pipe herzustellen, wird in einem Adressbucheintrag oder beim Aufbau einer Verbindung der Name der Pipe als Verbindungsziel angegeben.

Die Verbindung zu einer lokalen Pipe (d.h. einer Pipe auf dem lokalen Rechner) wird durch `\PIPE<name>` hergestellt, z.B. `\PIPE\OS2YOU`. Die Verbindung zu einer LAN-Server-Pipe wird durch `\\<server>\PIPE<name>` hergestellt, z.B. `\\ZAPHOD\PIPE\OS2YOU`.

1.8.8.2 Named Pipe Einstellungen

Wenn die Pipe im Datei-Menü auf 'Verbindungen annehmen' eingestellt ist, kann ZOC von Gegenstellen aus mit `\PIPE\ZOC` angerufen werden.

1.8.8.3 REXX ZocDeviceControl für Named-Pipe

`ZocDeviceControl` ist ein [ZOC-REXX Befehl](#), der es erlaubt, Funktionen, die spezifisch für eine

bestimmte Verbindungsmethode sind, auszuführen.

Es gibt keine Unterbefehle für den `ZocDeviceControl` Skriptbefehl in Bezug auf Named-Pipe Verbindungen.

1.8.9 RLogin

Rlogin ist eine Verbindungsmethode in ZOC, die [Telnet](#) sehr ähnlich ist, und dafür bestimmt ist, eine Remote-Shell über ein Netzwerk zu ermöglichen. Rlogin verwendet eine ungesicherte Verbindung, und wurde mittlerweile mehr oder weniger vollständig durch [SSH](#) ersetzt.

Die folgenden Themen behandeln die Verwendung von Rlogin in ZOC:

- ☐ [Rlogin Überblick](#)
- ☐ [Rlogin Einstellungen](#)
- ☐ [REXX ZocDeviceControl](#)

1.8.9.1 *Rlogin Überblick*

Der Verbindungstyp RLogin erlaubt es, per LAN oder Internet auf rlogin Server zuzugreifen. Um eine Verbindung zu einem RLogin Server aufzubauen, wird statt einer Telefonnummer entweder der Name des Hosts oder dessen IP-Adresse verwendet (z.B. im Adressbuch). Falls die Benutzer Einstellung leer ist, wird der Benutzername der Environmentvariable USERNAME entnommen. Falls diese nicht existiert, wird [ZOC](#) als Benutzername verwendet. Das Escape-Zeichen ist die Tilde "~".

1.8.9.2 *Rlogin Einstellungen*

Keine.

1.8.9.3 *REXX ZocDeviceControl für Rlogin*

`ZocDeviceControl` ist ein [ZOC-REXX Befehl](#), der es erlaubt, Funktionen, die spezifisch für eine bestimmte Verbindungsmethode sind, auszuführen.

Es gibt keine Unterbefehle für den `ZocDeviceControl` Skriptbefehl in Bezug auf Rlogin Verbindungen.

1.9 Die Programmierung von ZOC (REXX/DDE)

Es gibt zwei Methoden, ZOC zu automatisieren, nämlich REXX und DDE. Im folgenden finden Sie hierzu die entsprechenden Beschreibungen:

- ☐ [Einführung in die REXX Programmierung](#)
- ☐ [REXX-Sprachelemente \(Schleifen, Variablen, etc\)](#)
- ☐ [ZOC Befehle \(nach Funktionsgruppen\)](#)
- ☐ [ZOC Befehle \(alphabetisch\)](#)
- ☐ [Online REXX Dokumentation](#)
- ☐ [Übersicht Dynamic Data Exchange DDE \(Windows\)](#)
- ☐ [ZOC und AppleScript \(macOS\)](#)

1.9.1 Einführung in REXX

Über REXX und ZOC

Die von ZOC verwendete Skriptsprache REXX hat eine lange Tradition, speziell in der IBM-Welt. Die Sprache ist einfach aber vielseitig und bietet alle Elemente einer Programmiersprache, notwendig sind um Programme von einfacher bis mittlerer Komplexität zu schreiben (Variablen, Schleifen, Dateibehandlung, Prozeduren, usw.). Es gibt sogar eine objektorientierte Variante von REXX.

Die Programmierbarkeit von ZOC entsteht durch einer Kombination aus der Sprache REXX und ZOC-spezifischen Erweiterungen. Da REXX selbst bereits eine etablierte Skriptsprache ist, nutzt ZOC diese als Fundament und baut darauf mit eigenen Funktionen zur Terminalemulation und Kommunikation auf.

Diese Erweiterungen heben sich vom normalen Sprachumfang ab, indem Sie immer mit dem Namen `zoc` beginnen, z.B. `zocConnect`.

Hinweis: Die in ZOC verwendete REXX Interpreter ist REGINA REXX (<http://regina-rexx.sourceforge.net/>). Falls Sie eine andere REXX-Implementierung bevorzugen (z.B. OOREXX), können Sie diese unter [Optionen→Programmeinstellungen→Spezielle Dateien→Alternative REXX-DLL](#) konfigurieren.

ZOC REXX lernen

Nachfolgend finden Sie einen groben Einblick in die REXX Programmierung. Nachdem Sie diesen Überblick gelesen haben, sollten Sie einen Blick auf die Themen [REXX Sprachelemente](#) und [ZOC-REXX Befehle](#) werfen. Diese decken den notwendigen Sprachumfang für den größten Teil aller anfallenden Aufgaben ab.

Abgesehen davon befindet sich im Ordner [Eigene Dateien→ZOC9 Dateien→Rexx](#) das Archiv [ZocScriptingSamples.zip](#) mit deutschsprachigen Schritt-für-Schritt Beispielen. Darüber hinaus gibt es weitere [REXX Online Dokumentation](#) auf der EmTec-Webseite.

Sollten Sie REXX auf einem komplexeren Level verwenden wollen, sollten Sie den Text [ZOC REXX Reference \(PDF, engl\)](#) ansehen, den Sie auch im o.g. Ordner oder auf der Webseite finden.

Ein erstes Beispiel

Die ZOC-REXX Versionen des berühmten Hello-World Programms sehen so aus:

```
/* REXX */
-- "Hello World" am ZOC Bildschirm ausgeben
SAY "Hello World"
```

```
/* REXX */
-- "Hello World" zur Gegenstelle senden
CALL ZocSend "Hello World"
```

```
/* REXX */
-- Die Gegenstelle "Hello World" ausgeben lassen
CALL ZocSend "echo 'Hello World!'^M"
```

Grundsätzliches

Alle REXX-Programme beginnen in der Regel mit einem Kommentar. REXX sieht jeden Text, der zwischen `/*` und `*/` steht, also Kommentar an. Folglich sollte die erste Zeile eines REXX-Programms in etwa so aussehen: `/* REXX */`. Zusätzlich zu dieser Form von Kommentaren kennt REXX noch Kommentare bis zum Ende der Zeile, die durch `--` eingeleitet werden (siehe die Beispiele oben). REXX kennt Standard- und Erweiterungsbefehle. Beispiele für REXX Standardbefehle sind `SAY`, `IF`, `DO`, `END`. ZOC-Erweiterungsbefehle sind zum Beispiel `ZocConnect`, `ZocSend`, `ZocDisconnect`. Der Übersichtlichkeit halber werden die nativen REXX Sprachelemente in Großschrift verwendet, z.B. `CALL`, `IF`, `SAY`. ZOC-Erweiterungen werden dagegen in **CamelCase** angegeben, z.B. `ZocAsk`, `ZocWrite`. Namen die im Programm vom Benutzer selbst vergeben werden (z.B. Variablen, Prozeduren, etc.), stehen in Kleinbuchstaben, z.B. `ergebnis= 10*eingabe`.

Wie werden die verschiedenen Befehlstypen angesprochen

Standardbefehle

Standardbefehle werden durch Namen und ggf. Argumente angegeben (alle unten stehenden Wörter in Großbuchstaben sind Standard-REXX Befehle):

```
IF rc=640 THEN DO
    SAY "Anruf fehlgeschlagen!"
    SIGNAL done
END
ELSE DO
    SAY "Success!"
END
```

ZOC-Befehle

ZOC-Befehle werden als eine Art REXX-Unterfunktion aufgerufen und ausgeführt. Es gibt sie im Prinzip in zwei Formen:

Befehle, die keine Rückgabewert liefern (oder wenn Sie diesen nicht benötigen), werden durch `CALL` zusammen mit dem ZOC-Befehl und den benötigten, durch Komma getrennten, Argumenten aufgerufen. Befehle, die Rückgabewerte liefern, übergeben die Liste aller Argumente, in Klammern eingeschlossen, hinter dem Befehlsnamen.

```
/* Zwei Befehle ohne Rückgabewert: */
CALL ZocBeep 2
CALL ZocConnect "sss.hogwarts.edu"

/* Ein Befehl mit Rückgabewert: */
answer= ZocAsk("Wie ist Ihr Name")
```

Ein kleines Beispiel

Eine häufige Anwendung für REXX-Programme ist das Einloggen und die automatische Ausführung von Befehlen.

Einfache Login-Prozeduren können zwar auch ohne REXX Skript direkt im jeweiligen Eintrag des Adressbuchs als AutoLogin angegeben werden (siehe [Adressbucheinträge ändern](#)), deren Möglichkeiten sind jedoch auf sehr simple Abläufe begrenzt.

Das nachfolgende REXX Beispiel ist dagegen etwas komplexer. Es baut eine SSH Verbindung zu einem Rechner auf, führt dort den `mail` Befehl aus um zu sehen ob neue Email vorliegt, meldet sich wieder ab und zeigt ggf. eine Meldung.

```
/*REXX*/

-- variablen mit host und logininformationen
hostip= "users.hogwarts.edu"
username= "harryp"
password= "alohomora"

-- ssh verbindung aufbauen
CALL ZocSetDevice "Secure Shell"
CALL ZocConnect username":"password"@hostip

-- falls der rechner antwortet, sollten wir innerhalb
-- von wenigen sekunden den text "last login" sehen
-- (resultat von 640 beim wait-befehl bedeutet timeout)
CALL ZocTimeout 10
x= ZocWait("Last login")
IF x640 THEN DO
    -- erfolgreich angemeldet, auf prompt warten
    CALL ZocWait "$"

    -- sende den befehl 'mail'
    CALL ZocSend "mail^M"

    -- das echo des mail-befehls ueberspringen
    CALL ZocWaitline
```

```

-- die naechste zeile die kommt, ist die ausgabe
-- von 'mail' (z.b. 'no mail for harryp')
CALL ZocWaitline
themail= ZocLastline()

-- abmelden und verbindungsabbau
CALL ZocSend "exit^M"
CALL ZocDelay 1
CALL ZocDisconnect

/* wenn die antwort nicht mit "No mail" anfaengt,
   wird eine Meldung fuer den Benutzer angezeigt */
IF LEFT(themail,7)"No mail" THEN DO
    CALL ZocMsgBox themail
END
END

```

Hinweis: Weitere Beispiele finden Sie im [ZocScriptingSamples.zip](#) Archiv.

Was nun?

Nach diesem Überblick ist der nächste Schritt die Übersicht über die [REXX Sprachelemente](#).

1.9.2 Einführung in ZOC-DDE Programmierung (Windows)

Dynamic Data Exchange (DDE) erlaubt anderen Windows-Programmen, ZOC-Befehle ferngesteuert auszuführen. Während [REXX-Programme](#) unter der Kontrolle von ZOC laufen müssen, ist es mit DDE möglich, dass völlig separate Anwendungen (z.B. in C++ geschrieben) Befehle an ZOC zur Ausführung schicken.

In dieser Einführung wird davon ausgegangen, dass Kenntnisse über das Grundkonzept von DDE vorhanden sind. Die unten aufgeführten Punkte beschreiben nur Details der Implementierung. Eine detaillierte Beschreibung zusammen mit einem Beispielprogramm in C kann bei EmTec angefordert werden (<ftp://ftp.emtec.com/zoc/>).

Session/Topic

ZOC verarbeitet nur eine Session gleichzeitig. Der Application-Identifier ist `ZOC`, und ZOC antwortet auf die Topics `COMMUNICATION`, `COMM` und `COMM-DEBUG` (zweiterer zeigt zur Fehlersuche alle DDE-Events auf dem ZOC-Bildschirm an). ZOC antwortet nicht auf generelle DDE-Requests (also auf Requests ohne Application oder Topic).

Zusätzlich zum Namen `ZOC` reagiert ZOC auch noch auf `ZOCn`, wobei `n` eine Zahl ist, die sich aus der Anzahl der gleichzeitig gestarteten ZOC-Instanzen herleitet. Die erste heißt `ZOC1`, die nächste `ZOC2` usw.

Das Senden von Befehlen

Befehle können an ZOC entweder per `DDE_EXECUTE` (Datenfeld) oder `DDE_REQUEST` (Itemfeld) gesendet werden. `DDE_EXECUTE` antwortet mit einem `DDE_ACK` Paket. Der numerische Rückgabewert ist im Feld `AppRc` enthalten. `DDE_REQUEST` sendet immer ein `DDE_DATA` Paket (auch im Fehlerfall), welches das Ergebnis oder den Text `##ERROR##` im Datenfeld enthält. Für das `DDE_DATA` Paket darf kein `DDE_ACK` zurückgeschickt werden. ZOC für Windows erwartet, dass der Speicherblock, der die Daten enthält, vom Empfänger freigegeben wird.

Serialisierung

Befehle müssen streng auf Basis von Command-Reply gesendet werden. ZOC verarbeitet keine geschachtelten Befehle (z.B. das Senden eines zweiten Befehls, während noch ein ZocWait abgearbeitet wird).

Befehlsformat

DDE_EXECUTE und DDE_REQUEST verwenden das gleiche Befehlsformat. (Beide Befehle unterscheiden sich dadurch, dass EXECUTE nicht verwendet werden kann, um Rückgabewerte von Funktionen zu liefern). Die Befehle müssen mit Null abgeschlossen werden (wie Zeichenketten in C).

ZOC kann keine Befehle verarbeiten, die in eckigen Klammern eingeschlossen sind (wie in der Microsoft DDE Spezifikation empfohlen). Außerdem kann nur ein Befehl pro Message verarbeitet werden. Unicode wird nicht unterstützt.

Ein Befehl kann entweder in der Form `<befehl> <argumente>` oder in der Form `<befehl>(<argumente>)` übergeben werden. Die Argumente können mit Leerzeichen oder Komma getrennt werden. Die empfohlene Form sind allerdings Klammern und Komma. Zeichenketten sollten in Anführungszeichen gesetzt werden, es ist aber auch erlaubt, sie wegzulassen, wenn die Zeichenkette keine Spezialzeichen enthält.

Anbei eine Liste gültiger Zeichenketten. Um zu älteren ZOC-Versionen kompatibel zu sein, ist es erlaubt, den Text 'Zoc' am Anfang jedes Befehls wegzulassen.

```
ZocBeep(2)
ZocMsgBox("Hello World!", 2)
ZocNotify('Datei "FOOBAR.DATA" nicht gefunden!')

ZocRequest "Wie geht's?" "Gut" "Nicht schlecht"
Notify Error
Beep 2
```

Liste der Befehle

Per DDE kann der gleiche Befehls-/Funktionsumfang wie in REXX verwendet werden. Eine Auflistung aller Befehle ist unter [ZOC-Befehle](#) im Anhang zu finden.

Beispiel

Der folgende Ausschnitt ist ein kleines Microsoft Access VBA Programm, das eine DDE-Verbindung zu ZOC aufbaut, das Modem initialisiert, und auf "OK" wartet.

```
ZocDDE = DDEInitiate("ZOC", "Comm-Debug")
DDEExecute ZocDDE, "ZocClearScreen"
DDEExecute ZocDDE, "ZocTimeout(10)"
DDEExecute ZocDDE, "ZocSend('ATZ^M')"
rc = DDERequest(ZocDDE, "ZocWait('OK')")
' oder ab Excel 2003:
'   rlist= DDERequest(ZocDDE, "ZocWait('OK')")
'   rc= rlist(1)
Debug.Print rc
DDETerminate ZocDDE
```

1.9.3 Einführung in ZOC-AppleScript Programmierung (macOS)

ZOC bietet ein simples Interface, um [ZOC-Befehle](#) von AppleScript aus auszuführen.

Um ZOC auf die Befehls-Ausführung durch AppleScript vorzubereiten, müssen die auszuführenden Befehle zwischen `start conversation` und `end conversation` eingeschlossen werden.

Aufruf von ZOC-Befehlen als Unterfunktion

ZOC-Befehle werden in AppleScript genauso verarbeitet, als wenn sie von innerhalb der ZOC-Skriptsprache (REXX) aus aufgerufen worden wären, und auch die Syntax ist ähnlich. Um einen Befehl auszuführen, muss dieser einfach mit Funktionsaufruf-Syntax angegeben werden (die gleiche Syntax, die auch für den Aufruf von Subroutinen innerhalb eines AppleScripts verwendet wird).

Ein einfaches AppleScript mit einem ZOC Funktionsaufruf sieht folglich so aus:

```
tell application "zoc9"
    -- bereite die aktuelle Verbindung im ersten ZOC-Fenster auf die
    -- Verarbeitung von ZOC-Befehlen vor
    start conversation

    -- sende 'ls -l<enter>' zur Gegenstelle
    ZocSend("ls -l^M")

    -- öffne in ZOC ein Abfragefenster und speichere die Auswahl
    set reply to ZocRequest("Ihre Auswahl?", "Äpfel", "Orangen")

    -- gebe die Befehls-Verarbeitung wieder frei
    end conversation
end tell
```

In der [ZOC Funktionsliste](#) finden Sie eine Beschreibung sämtlicher Funktionen und Parameter (diese Liste kann auch direkt aus dem Hilfemenü von ZOC aufgerufen werden). Die Syntax der Befehle und die dort angegebenen Beispiele beziehen sich auf den in ZOC verwendeten internen REXX Skript-Interpreter, sollten aber zusammen mit dem obigen Beispiel selbsterklärend sein.

Aufruf von ZOC-Befehlen mit 'perform'

Wenn Sie eine Syntax bevorzugen, die mehr an AppleScript angelehnt ist, können Sie ZOC-Befehle auch durch den `perform` Befehl ausführen. Dieser Befehl erhält als direkten Parameter den Namen des ZOC-Befehls, sowie ein Array (in geschweiften Klammern), das alle Parameter für diesen Befehl enthält.

Diese Variante ist funktional identisch zum Aufruf als Unterfunktion. Es stellt lediglich eine Alternative dar:

```
tell application "zoc9"
    -- bereite die aktuelle Verbindung im ersten ZOC-Fenster auf die
    -- Verarbeitung von ZOC-Befehlen vor
    start conversation

    -- sende 'ls -l<enter>' zur Gegenstelle
    perform "ZocSend" using { "ls -l^M" }

    -- öffne in ZOC ein Abfragefenster und speichere die Auswahl
    set reply to perform "ZocRequest" using
        { "Ihre Auswahl?", "Äpfel", "Orangen" }

    -- gebe die Befehls-Verarbeitung wieder frei
    end conversation
end tell
```

1.9.4 REXX Sprachelemente

Diese Übersicht behandelt spezielle Themen über die Kontrollsprache REXX (alternativ ist eine allgemeiner gehaltene ZOC REXX-Einführung unter [REXX Programmierung](#) zu finden).

Die REXX Sprache für ZOC besteht im Prinzip aus drei Teilen:

1. REXX Grundbefehle (*IF*, *LOOP*, *CALL*, usw.)
2. REXX Standardfunktionen. (*TIME()*, *SUBSTR()*, *STRIP()*, usw.)
3. ZOC Erweiterungen (*ZocSend*, *ZocDownload*, *ZocWait*, etc.)

Der Text unten beschreibt den Grundumfang (Punkte 1 und 2). Die speziellen ZOC-Erweiterungen (Punkt 3) sind unter [Hilfe-Menü→ZOC REXX-Befehle/Funktionen](#) zu finden.

Neben der Beschreibung hier, ist der komplette REXX Grundumfang (wiederum Punkte 1. und 2.) auch an verschiedenen Stellen im Internet beschrieben. Links hierzu befinden sich im [Rexx-Ordner](#) innerhalb des Ordners [ZOC Dateien](#) (wir empfehlen das REXX Wiki (deutsch)).

Die [ZOC REXX Reference \(PDF, engl\)](#) deckt alle drei Punkte ab. Die unten aufgeführten Themen werden außerdem in den REXX-Beispielen gezeigt, die über den REXX Link im o.g. Ordner bzw. in der [REXX Online Dokumentation](#) verfügbar sind.

Programm-Formatierung

Alle REXX-Programme müssen prinzipiell mit einem Kommentar beginnen. Als Kommentar gilt Text, der zwischen */** und **/* steht. Außerdem gibt es Zeilenkommentare, die durch *--* eingeleitet werden und bis zum Ende der Zeile gelten.

Es können mehrere Befehle in eine Zeile geschrieben werden, wenn diese durch Komma voneinander getrennt werden. Ein Befehl kann in der nächsten Zeile fortgesetzt werden, wenn am Ende der vorherigen Zeile ein zusätzliches Komma geschrieben wird.

```
/* REXX */

SAY "Hello "; SAY "World"
CALL zoc "REQUEST", "Wie geht's", ,
    "Gut", "Nicht schlecht"
answer= ZOCRESULT()
-- jetzt ist genug
EXIT
```

Zahlen und Berechnungen

Zahlen können zum Rechnen, Zählen usw. verwendet werden. REXX verwendet Zahlen und Berechnungen in einer denkbar unkomplizierten Weise:

```
/* REXX */
SAY 4*5+3*5 /* Ausgabe: 35 */
SAY 10/3     /* Ausgabe: 3.333333 */
SAY 10%3     /* Ausgabe: 3 (Ganzzahl-Division) */
SAY 10//3    /* Ausgabe: 1 (Divisions-Rest) */
N= 4
SAY N        /* Ausgabe: 4 */
N= N+1
```

```
SAY N*3      /* Ausgabe: 15 */
```

Zeichenketten

Obwohl REXX in der Lage ist, mit Text ohne Anführungszeichen zu arbeiten, ist es generell besser, Zeichenketten mit einfachen oder doppelten Anführungszeichen zu versehen:

```
/* REXX */
SAY "Hello World!"
SAY "Joe's Bar"
SAY 'Er sagt: "Verwende ZOC!'"
SAY 'Sie sagt: "Mach ich sowieso schon!"'
```

Zeichenketten können Variablen zugewiesen werden. Außerdem können Zeichenketten und Variablen direkt aneinandergereiht werden. Falls syntaktisch notwendig kann auch der String-Verbindungsoperator `||` verwendet werden:

```
/* REXX */
w= "World"
ex= "!"
w2= w||ex  -- the || operator is actually required here (w||ex vs. wex)
hwe= "Hello "w2

-- fuenf mal "Hello World!"
SAY hwe
SAY "Hello" w2
SAY "Hello "w2
SAY "Hello "||w2
SAY "Hello "||w||ex
```

Ablaufverfolgung

Zur Fehlersuche ist es möglich, am Anfang des Programms nach der ersten Zeile den Befehl `TRACE A` anzugeben. Danach zeigt der REXX-Interpreter jeden Befehlsschritt vor der Ausführung an. `TRACE I` zeigt eine sehr detaillierte Ablaufverfolgung, `TRACE O` schaltet die Ablaufverfolgung wieder aus.

Vergleiche

Um Vergleiche in REXX auszudrücken, stehen folgende Vergleichs-Operatoren zur Verfügung:

Operator	Funktion	Beispiel
=	Numerische Gleichheit	IF ret=640 THEN ...
==	Gleichheit bei Zeichenketten	IF input=="ZOC" THEN ...
<>	nicht gleich	IF rc<>640 THEN ...
\=	nicht gleich (wie <>)	IF rc\=640 THEN ...

<	kleiner als	IF val<10 THEN ...
>	größer als	IF val>50 THEN ...
<=	kleiner oder gleich	IF val<=100 THEN ...
>=	Größer oder Gleich	IF val>=100 THEN ...
&	UND-Verknüpfung	IF (i>=0) & (i<10) THEN ...
	ODER-Verknüpfung	IF (k=0) (k=1) THEN ...
\	NICHT-Verknüpfung	IF \ (k=0) THEN ...

Entscheidungen

Entscheidungen werden in der Syntax *IF <ausdruck> THEN DO <befehle> END ELSE DO <befehle> END* ausgedrückt (siehe Abschnitt Vergleiche (oben) für die Vergleichs- und Verknüpfungsoperationen).

Ein typischer IF-Programmteil sieht in etwa so aus:

```
/* REXX */
IF rc=0 | rc=1 THEN DO /* rc 0 OR rc 1 */
    SAY "ok"
END
ELSE DO
    SAY "failed"
END
```

Der ELSE-Zweig kann weggelassen werden, wenn er nicht benötigt wird. Die Schlüsselwörter *DO* und *END* können weggelassen werden, wenn sich nur ein Befehl zwischen DO und END befindet.

```
/* REXX */
IF rc=0 | rc=1 THEN SAY "ok"
ELSE SAY "failed"
```

Schleifen

Schleifen werden in REXX folgendermaßen formuliert:

```
DO <zähler> <befehle> END
DO WHILE <bedingung> <befehle> END
DO UNTIL <bedingung> <befehle> END
DO <variable>=<start> TO <ende> <befehle> END
```

```

/* 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

```

Schleifen können mit dem Befehl `LEAVE` abgebrochen werden. Mit `ITERATE` kann innerhalb der Schleife der nächste Schleifendurchgang ausgelöst werden (d.h. der Befehl überspringt den Rest des aktuellen Schleifendurchgangs).

Sprünge und Prozeduren

Sprungmarken und Prozeduren werden mit einem Namen und nachfolgendem Doppelpunkt gekennzeichnet. Um eine Sprungmarke anzuspringen, wird der Befehl `SIGNAL` verwendet, während der Befehl `CALL` eine Prozedur aufruft (die Prozedur kann zum Aufrufpunkt mittels `RETURN` zurückkehren).

Das nachfolgende Beispiel zeigt einen Sprung und einen Prozeduraufruf mit Argumentübergabe:

```

/* 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 "Das Quadrat von "||value||" ist "||sqr
    RETURN

```

Funktionsaufrufe

Funktionen werden wie Prozeduren gekennzeichnet und aufgerufen. Allerdings besitzen Funktionen im Gegensatz zu Prozeduren einen Rückgabewert:

```

/* REXX */

SAY "Basis eingeben"
PULL b
SAY "Exponent eingeben"
PULL p

result= power(b,p)
SAY "Der " p "te Exponent von " b " ist " result

EXIT

power:
    base= ARG(1)
    pow= ARG(2)
    res= 1
    DO I=1 TO pow
        res= res*base
    END
    RETURN res

```

Externe Skripts

Es ist auch möglich ein externes Skript als Unterprogramm oder Funktion aufzurufen. Dazu muss der Name der Datei in Hochkommata geschrieben werden und die Datei entweder im ZOC-Programmordner, im ZOC Dateiordner oder im ZOC Skript-Ordner oder in einem Ordner im Windows Suchpfad (PATH) oder im REGINA REXX Makro Pfad (REGINA_MACROS) liegen. Alternativ kann die Datei komplettem mit Pfad angegeben werden.

Parameter werden wie sonst auch mit Komma getrennt aufgelistet und können im externen Skript mittels der [ARG\(n\)](#) Funktion abgegriffen werden.

```

/* REXX:test.zrx */

/* externes Skript in einem der ZOC ordner */
CALL "sub.zrx" "Hello", "World"

/* externes Skript in einem anderen Ordner */
CALL "C:\somefolder\sub.zrx" "Hello", "World"

/* externes Skript als Funktion */
x= "sub.zrx"(12,2)
SAY x

/* externes Skript per INTERPRET */
scriptfile= "Z:\test\zzz.zrx"
cmd= 'CALL "'||scriptfile||'"'
INTERPRET cmd

/* externes Skript als Funktion per INTERPRET */
scriptfile= "sub.zrx"
cmd= 'x= "'||scriptfile'"(12,2)'
INTERPRET cmd
SAY x

```

Das aufgerufene Skript kann die Parameter durch die REXX Funktion `ARG(n)` aufnehmen und Ergebnisse per `RETURN` zurückgeben:

```
/* REXX:sub.zrx */
a= ARG(1)
b= ARG(2)
SAY a
SAY b
mult= a*b
RETURN mult
```

Der PARSE Befehl

PARSE ist ein flexibles REXX Kommando um formatierte Zeichen in Teile zu zerlegen und die Teile Variablen zuzuweisen. Der Syntax hierfür ist `PARSE VALUE <zeichenkette> WITH <variable>"<trenner>" ...`. Wenn Sie z.B. eine Zeichenkette coord mit Koordinaten in der Form `<index>: <pos-x>/<pos-y>` haben, können Sie diese leicht mittels

```
PARSE VALUE coord WITH index": "posx"/"posy zerlegen.
```

Beispiel:

```
/* REXX BEISPIEL UM EIN MIT KOMMA GETRENNTE LISTE VON ZAHLEN ZU ZERLEGEN */
x= "1, 2, 3, 4"

DO FOREVER
    IF x==" " THEN LEAVE

    PARSE VALUE x WITH zahl", "rest

    SAY zahl

    x= rest
END
```

Verwendung von ZOC-Befehlen

ZOC-Befehle sind eine Erweiterung der REXX Sprache in Form von Prozeduren und Funktionen und bieten Zugriff auf ZOC-spezifische Funktionalitäten (siehe [ZOC-REXX Befehle](#)).

ZOC-Befehle, die keinen Rückgabewert besitzen (oder wenn Sie an dem Rückgabewert nicht interessiert sind), werden wie Prozeduren aufgerufen:

```
CALL <zoc-befehlsname> <argument(e)>
```

ZOC-Befehle, die einen Rückgabewert besitzen, werden wie Funktionen aufgerufen:

```
<ergebnis>= <zoc-befehlsname>(<argument(e)>)
```

Verwendung von macOS- oder Windows-Befehlen

Befehle, die direkt vom Betriebssystem ausgeführt werden (wie z.B. Löschen und Umbenennen von Dateien), müssen direkt an das Betriebssystem adressiert werden. Dies kann entweder durch REXX's `ADDRESS CMD` Befehl oder durch ZOC's `ZocShell` Befehl geschehen.

```

/* REXX */
ADDRESS CMD "cmd.exe /c DEL UPLOAD.TMP"
CALL ZocShell "DEL UPLOAD.TMP"

```

Eine hilfreiche Funktionalität von `ADDRESS CMD` ist die Umleitung der Ausgabe in eine STEM (Array) Variable:

```

/* REXX */
ADDRESS CMD "CMD /C DIR" WITH OUTPUT STEM out.
SAY out.0 "Ausgabezeilen"
DO i=1 TO out.0
    SAY out.i
END

```

Eingebaute Funktionen

Die nachfolgenden Funktionen können in Zuweisungen oder in anderen Befehlen, die Werte erwarten, verwendet werden, z.B. `b= ABS(a)` oder `IF ABS(n)>10 THEN ....`

.Es sind nur die wichtigsten Funktionen und Argumente aufgelistet. Der komplette Sprachumfang ist in der [ZOC REXX Reference \(PDF, engl\)](#) beschrieben, die sich im Ordner [Eigene Dateien→ZOC9 Dateien→Rexx](#) befindet.

ABS(<wert>)

Absoluter Wert (entfernt Vorzeichen), z.B. `n=ABS(t)`

ARG(<n>)

Liefert das n-te Argument einer Prozedur/Funktion.

COPIES(<str>, <anzahl>)

Gibt <anzahl> Kopien von <string> zurück, z.B. `x=COPIES("-mehr- ", 10)`

C2D(<zeichen>)

Liefert den dezimalen ASCII-Wert eines oder mehrerer Zeichen (Char-TO-Decimal), z.B. `n=C2D('A')` setzt n auf 65.

C2X(<zeichen>)

Liefert den dezimalen ASCII-Wert eines Zeichens (Char-TO-heX), z.B. `n= C2X('AB')` setzt n auf 4142.

DATE("<format>")

Gibt das aktuelle Datum in verschiedenen Formaten zurück (eines von B, D, E, M, N, O, S, U, W), z.B. `heute=DATE("S")` gibt am 1.2.97 den Text "19970201" zurück.

D2C(<ascii>)

Liefert das Zeichen zu einem dezimalen ASCII-Wert (Decimal-To-Character), z.B. `SAY D2C(65)` gibt den Buchstaben A aus, bzw. `cr= D2C(13)` weist der Variable CR das Carriage Return Zeichen zu.

FILESPEC(<teil>, <dateiname>)

Liefert einen Teil des Dateinamens (aus "Drive", "Path", "Name"), z.B. `dir=FILESPEC("Path", ofile)`

LEFT(<string>, <num>)

Liefert die ersten <num> Zeichen von <string>, z.B. `SAY LEFT("BIERFLASCHE", 4)` gibt BIER aus.

LENGTH(<string>)

Gibt die Anzahl Zeichen des Strings zurück.

LINEIN(<dateiname>)

Liest die nächste Textzeile einer Datei aus (siehe Datei-I/O).

LINEOUT(<dateiname>, <text>)

Schreibt die nächste Textzeile in eine Datei (siehe Datei-I/O).

POS(<string1>, <string2>)

Sucht den ersten String im zweiten String und liefert die Position oder 0, wenn nicht gefunden.

RIGHT(<string>, <num>)

Liefert die letzten <num> Zeichen von <string>, z.B. `SAY RIGHT("BIERFLASCHE", 7)` gibt FLASCHE aus.

STREAM(<dateiname>, ...)

Führt eine Dateioperation aus (siehe Datei-I/O).

SUBSTR(<string>, <pos>[, <länge>])

Liefert <länge> Zeichen ab Position <pos> in <string> zurück. Wird <länge> nicht angegeben, wird der Rest von <string> zurückgeliefert.

STRIP(<string>)

Entfernt Leerzeichen am Anfang und Ende einer Zeichenkette, z.B. `str= STRIP(str)`

TIME("<format>")

Gibt die aktuelle Uhrzeit von verschiedenen Formaten (eines von C, H, L, M, N, S) zurück, z.B. `SAY TIME("N")` zeigt die Uhrzeit in der Form HH:MM:SS.

TRANSLATE(<string>)

Setzt alle Zeichen von <string> in Großbuchstaben um.

TRUNC(<n>, <m>)

Verkürzt den Wert <n> auf <m> Dezimalstellen.

X2C(<hex>)

Liefert das Zeichen zu einem hexadezimalen ASCII-Wert (heX-To-Character), z.B. `SAY X2C(41)` gibt den Buchstaben A aus, bzw. `CrLf= X2C(0D0A)` weist der Variable CRLF die zwei Zeichen für Carriage-Return/Line-Feed zu.

Datei-I/O

Überprüfen, ob eine Datei existiert

`IF STREAM(<dateiname>, "C", "QUERY EXISTS")\="" THEN ...`

Zum Schreiben öffnen

`CALL STREAM <dateiname>, "C", "OPEN WRITE"`

Zum Lesen öffnen

`CALL STREAM <dateiname>, "C", "OPEN READ"`

In eine Datei schreiben

`CALL LINEOUT <dateiname>, <text>`

Aus einer Datei lesen

```
<variable>=LINEIN(<dateiname>)
```

Auf EOF abprüfen

```
IF STREAM(<dateiname>, "S")\="READY" THEN ...
```

Eine Datei schließen

```
CALL STREAM <dateiname>, "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"
```

Ein hilfreicher Trick zum Lesen einer Textdatei ist die Verwendung von [ADDRESS CMD](#) bei gleichzeitiger Umleitung der Ausgang in eine STEM (Array) Variable:

```
/* 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
```

Die Datei `FILEIO.ZRX` bei den Beispielen im REXX Ordner wendet diese Funktionen beispielhaft an.

Übergabe von Parametern an REXX

Von der ZOC Kommandozeile bzw. von ZOC-Benutzerknöpfen kann auch ein Parameter an ein REXX Skript übergeben werden, z.B. `ZOC "/RUN:script\test.zrx Hello, World"`

Solche Parameter können in im REXX Skript mit Hilfe der ARG(1) Funktion abgefragt und ggf. mit `PARSE` zerlegt werden) (es wird dabei noch dringend empfohlen evtl. führende und nachfolgende Leerzeichen durch `STRIP()` zu entfernen).

```
/* REXX PARAMETERÜBERGABE */
p= ARG(1)
PARSE VALUE p WITH p1", "p2
p1= STRIP(p1)
p2= STRIP(p2)
SAY "Übergebene Parameter: ("||p||") ("||p1||") ("||p2||")"
```

Häufige Funktionen in Visual Basic vs. REXX

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#
CALL LINEOUT <file>, ...

1.9.5 ZOC-REXX Funktionen nach Funktionsgruppen

Unten finden Sie eine Liste der ZOC-Befehle/Funktionen nach Funktionsgruppen geordnet. Eine komplette [alphabetische ZOC Funktionsliste](#) befindet sich dagegen im Anhang. Außerdem gibt es eine Beschreibung der REXX-Grundbefehle wie Variablen, Schleifen, usw. in [REXX Sprachelemente](#).

Verbindung, Senden, Übertragen

- ☐ [ZocConnect](#)
- ☐ [ZocConnectHostdirEntry](#)

- ☐ [ZocDisconnect](#)
- ☐ [ZocDownload](#)
- ☐ [ZocLastLine](#)
- ☐ [ZocPing](#)
- ☐ [ZocReceiveBuf](#)
- ☐ [ZocRespond](#)
- ☐ [ZocSend](#)
- ☐ [ZocSendEmulationKey](#)
- ☐ [ZocSendRaw](#)
- ☐ [ZocSetAutoAccept](#)
- ☐ [ZocTimeout](#)
- ☐ [ZocUpload](#)
- ☐ [ZocWait](#)
- ☐ [ZocWaitForSeq](#)
- ☐ [ZocWaitIdle](#)
- ☐ [ZocWaitLine](#)
- ☐ [ZocWaitMux](#)

Ein/Ausgabe im Terminal und in Fenstern

- ☐ [ZocAsk](#)
- ☐ [ZocAskPassword](#)
- ☐ [ZocAskFilename](#)
- ☐ [ZocAskFileNames](#)
- ☐ [ZocAskFoldername](#)
- ☐ [ZocBeep](#)
- ☐ [ZocClearScreen](#)
- ☐ [ZocDialog.](#)
- ☐ [ZocGetScreen](#)
- ☐ [ZocKeyboard](#)
- ☐ [ZocMessageBox](#)
- ☐ [ZocNotify](#)
- ☐ [ZocPlaySound](#)
- ☐ [ZocRequest](#)
- ☐ [ZocRequestList](#)
- ☐ [ZocSuppressOutput](#)
- ☐ [ZocWrite](#)
- ☐ [ZocWriteln](#)
- ☐ [ZocWindowState](#)

Einstellungen auslesen/ändern

- ☐ [ZocGetProgramOption](#)
- ☐ [ZocGetSessionOption](#)
- ☐ [ZocGetHostEntry](#)
- ☐ [ZocLoadKeyboardProfile](#)
- ☐ [ZocLoadSessionProfile](#)
- ☐ [ZocLoadTranslationProfile](#)
- ☐ [ZocSaveSessionProfile](#)
- ☐ [ZocSetDevice](#)
- ☐ [ZocSetDeviceOpts](#)
- ☐ [ZocSetEmulation](#)
- ☐ [ZocSetHostEntry](#)
- ☐ [ZocSetProgramOption](#)
- ☐ [ZocSetSessionOption](#)
- ☐ [ZocSetUnattended](#)

Manipulation von Dateien, Ausführen von Programmen

- ☐ [ZocFilename](#)
- ☐ [ZocFileCopy](#)
- ☐ [ZocFileDelete](#)
- ☐ [ZocFileRename](#)
- ☐ [ZocListFiles](#)
- ☐ [ZocShell](#)
- ☐ [ZocShellExec](#)
- ☐ [ZocShellOpen](#)

Operationen in ZOC durchführen

- ☐ [ZocCommand](#)
- ☐ [ZocDelay](#)
- ☐ [ZocGetInfo](#)
- ☐ [ZocLogname](#)
- ☐ [ZocLogging](#)
- ☐ [ZocMenuEvent](#)
- ☐ [ZocSessionTab](#)
- ☐ [ZocTerminate](#)

Hilfsfunktionen

- ☐ [ZocClipboard](#)
- ☐ [ZocString](#)
- ☐ [ZocMath](#)
- ☐ [ZocGlobalStore](#)
- ☐ [ZocRegistry](#)

Komplexe, subtile und obskure Funktionen

- ☐ [ZocCtrlString](#)
- ☐ [ZocDdeClient](#)
- ☐ [ZocDoString](#)
- ☐ [ZocDeviceControl](#)
- ☐ [ZocEventSemaphore](#)
- ☐ [ZocSetAuditLogname](#)
- ☐ [ZocSetMode](#)
- ☐ [ZocSetTimer](#)
- ☐ [ZocSynctime](#)

1.9.6 Alphabetische Liste der ZOC-Funktionen

Eine alphabetische Liste der ZOC-Funktionen finden Sie unter [ZOC-REXX Befehle](#) im Anhang.

1.10 Updates, Autor, Email, Bestellung, ...

1.10.1 Kontakt mit EmTec

Bestellinformationen

Falls Sie Fragen zu Preisen oder Bestellungen haben, wählen Sie bitte den Punkt [Bestellen](#) im Hilfe-Menü. Dort finden Sie alle Händler und deren Kontaktadressen.

Support

Falls Sie Probleme mit dem Programm haben, werden wir versuchen Ihnen zu helfen, aber lesen Sie bitte den Abschnitt [Häufige Fragen und Antworten](#) bevor Sie uns kontaktieren

Wenn Sie dort keine Antwort finden, können Sie uns gerne über den Support-Bereich unserer Webseiten erreichen: <http://www.emtec.com/common/support.htm>

Kontakt

Bei komplexen technischen Problemen, wegen Entwicklungsunterstützung, Spezialanpassungen, Fehlermeldungen, Firmenlizenzen, Ideen oder falls Sie uns Weihnachtsgeschenke schicken wollen:

<http://www.emtec.com/common/contact.htm> oder

EmTec

Markus Schmidt

Kirchenweg 14

90419 Nürnberg

- Deutschland -

1.10.2 Wo gibt es Updates

<http://www.emtec.com/zoc/index.htm>

<http://www.bmtmicro.com>

<ftp://ftp.emtec.com/zoc>

1.10.3 Copyright und Warenzeichen

ZOC ist (C) 1993-2025 by

EmTec, Innovative Software

Markus Schmidt

<http://www.emtec.com>

ZOC und EmTec sind registrierte Warenzeichen und U.S.-Trademarks von Markus Schmidt

Diese Software enthält außerdem Programmcode und Programmteile (im Einklang mit deren jeweiligen Lizenzbestimmungen) von folgenden Quellen:

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. 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>

ZOC REXX/REGINA REXX

Der ZOC REXX Interpreter basiert auf REGINA REXX (C) 2008 Mark Hessling (<http://regina-rexx.sourceforge.net/>).

Entsprechend der LGPL Lizenz (siehe ZOCCREXX-COPYING.TXT) von REGINA REXX können Sie den kompletten Sourcecode für ZOCCREXX.DLL über unseren Support bekommen.

OOREXX

Das OOREXX System ist (C) by www.oorexx.org und wird unter der OOREXX Lizenz verwendet: <http://www.oorexx.org/license.html>

InfoZip/Install

Copyright (C) 1990-1992 Mark Adler, Richard B. Wales, Jean-loup Gailly, Kai Uwe Rommel und Igor Mandrichenko
explode.c -- Not copyrighted 1992 by Mark Adler
inflate.c -- Not copyrighted 1992-94 by Mark Adler

TN3270

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.
<http://x3270.bgp.nu/x3270-man.html#Copyrights>

CRC Routinen

UPDCRC Makro abgeleitet aus einem Artikel Copyright (C) 1986 Stephen Satchell
CRC-Tabelle 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 und Mark Adler

Zmodem

Original public domain version by Chuck Forsberg, Omen Technology Inc. (C) 1988

Trademarks

ZOC, EmTec

ZOC und EmTec sind eingetragene Warenzeichen von Markus Schmidt

Microsoft, Microsoft Windows

Microsoft und Microsoft Windows sind eingetragene Warenzeichen der Microsoft Corporation

VT52, VT100, VT102, VT220, VT320, VT420, VT520

VT52, VT100, VT102, VT220, VT320, VT420 und VT520 sind eingetragene Warenzeichen der Digital Equipment Corporation

IBM, IBM3270, IBM5250, AS/400, OS/2, iSeries, eServer

IBM, IBM3270, IBM5250, AS/400, OS/2, iSeries und eServer sind eingetragene Warenzeichen der IBM Corporation

Apple, Macintosh, macOS

Apple, Macintosh, macOS sind eingetragene Warenzeichen der Apple Inc.

1.11 ANHANG

- ☐ [VT102/VT220 Sondertasten auf der PC Tastatur](#)
- ☐ [3270 Sondertasten auf der PC Tastatur](#)
- ☐ [Tastenbezeichnung für die Funktion ZocSendEmuKey/^KEY](#)
- ☐ [Tabelle der Steuerzeichen und Aktionscodes](#)
- ☐ [Erweiterte Escape-Sequenzen](#)
- ☐ [ZOC REXX-Befehle/Funktionen](#)
- ☐ ['Wie geht das?'-Fragen](#)
- ☐ [Leicht zu übersehende Funktionen](#)
- ☐ [Häufige Probleme, Fragen und Antworten](#)

1.11.1 Steuerzeichen, Platzhalter- und Aktionscodes

- ☐ Steuerzeichen
- ☐ Platzhalter-Codes

Oftmals ist es notwendig, Zeichen zu senden, die nicht auf der Tastatur verfügbar sind, bzw. die nicht in ein Eingabefeld eingegeben werden können, weil das Betriebssystem diese Tasten zum Steuern der Eingabefelder verwendet (z.B. Esc, Tab oder Return).

Falls Sie Steuerzeichen per Tastatur vom Terminalfenster aus senden müssen, z.B. ^] um einen Cisco-Router anzusprechen, finden Sie die entsprechende Tabelle bei den [Listen der Tastaturabkürzungen](#).

Um stattdessen solche Zeichen in Eingabefeldern (z.B. beim Belegen von Benutzerknöpfen) auch eingeben zu können, verwendet ZOC deshalb Steuerzeichen, die dann intern durch die entsprechenden Tasten ersetzt werden.

ZOC verwendet außerdem spezielle [Platzhaltercodes](#), die es erlauben, Werte wie das aktuelle Datum oder die aktuelle Uhrzeit in Dateinamen zu verwenden.

Steuerzeichen

Die hier gelisteten Codes können in Eingabefeldern verwendet werden um Spezialtasten zu simulieren, die anders nicht in im Dialog eingegeben werden können (z.B. bei der Tastenbelegung). Wenn Sie dort zum Beispiel `ZOC<Enter>` senden möchten, geben Sie stattdessen `ZOC^M` ein.

Code	Beschreibung	Emulationstaste Äquivalent	TN3270-Taste Äquivalent
^M	Carriage Return (CR, hex 0D)	Enter	Newline
^[	Escape \\E	Esc	 .

	\033		
^H	Backspace (hex 08)	Backspace	Backspace
^I	Tabulator (hex 09)	Tab	Tab
^J	Zeilenvorschub (LF)	 	Newline
^Z	Suspend process (hex 1A)	Ctrl+Z	Datenfreigabe/Enter
Andere ^A ... ^Z	Kombination mit Strg-Taste	Strg+A ... Strg+Z	
^~	Strg+^ (dec 30)	 	
^[	DEC Control Squence Introducer	 	
	CSI		
^^	^-Zeichen	 	
^?	DEL Zeichen (127 bzw. hex 7F)	 	
^!	Sendet ein Modem-Break, Telnet-Break, etc.	 	
^°	Ca 125ms Pause vor dem Senden des nächsten Zeichens	 	
^(xx)	Beliebiges Zeichen Hex-code z.B. ^ (7F) für DEL	 	

Platzhalter Codes

Die Tabelle unten listet Platzhalter, die verwendet werden können um in Felder Werte wie Datum, Uhrzeit und verbindungsbezogene Informationen einzufügen. Die Codes können überall verwendet werden, wo auch andere Steuerzeichen akzeptiert werden. Außerdem ist die Verwendung in Namen für Logdateien möglich (siehe [Optionen→Verbindungsprofil→Log](#)).

Code	Wert/Ersetzung
^%	Wird durch den im gerade aktiven Adressbucheintrag angegebenen Benutzernamen ersetzt
^&	Wird durch das im gerade aktiven Adressbucheintrag angegebene Kennwort ersetzt (aus Sicherheitsgründen funktioniert dieser Code nach jedem Login nur einmal).
^+	Wird durch den Namen der Gegenstelle ersetzt.
^1	Wird durch den aktuellen Tag ersetzt
^2	Wird durch das aktuelle Monat ersetzt
^3	Wird durch das aktuelle Jahr ersetzt
^4	Wird durch die momentane Stunde ersetzt
^5	Wird durch die momentane Minute ersetzt
^6	Wird durch die momentane Sekunde ersetzt
^7	Wird durch den Namen des Computers ersetzt
^8	Die laufende Nummer des aktuell offenen ZOC Fensters. Falls das Fenster Registerblätter enthält, wird zusätzlich eine Nummer für das Register hinzugefügt. (Dieser Code dient primär dazu, eindeutige Dateinamen erzeugen zu können.)
^9	ZOC Version und Betriebssystem, z.B. ZOC9.02.7-WIN10

1.11.2 Erweiterte Escape-Sequenzen

Die folgenden Escape-Sequenzen erweitern den normalen Bereich von Sequenzen (die normalen Sequenzen definieren das Verhalten von ZOC je nach der gewählten Emulation).

Der Hauptzweck der erweiterten Sequenzen besteht darin, einem Host zu ermöglichen, bestimmte Funktionen zu steuern oder bestimmte Aktionen über ZOC aufzurufen.

Die Sequenzen beginnen in der Regel mit `ESC ]` (hex 1B 5D) oder `ESC [` (hex 1B 5B), gefolgt von

einem variablen Teil und einem Abschlusszeichen wie `^G` (*hex 07*) oder `CR` (*hex 0D*).

Andere Zeichen in den syntaktischen Beispielen unten sind selbsterklärend, abgesehen von Leerzeichen (die ignoriert werden sollten) und Text in Klammern (der als Platzhalter für einen variablen Teil deiner eigenen Wahl dient).

Beispielsweise ist eine gültige Implementierung der Sequenz `ESC ] 0 ; <title> ^G` die Byte-Folge *hex 1B 5D 30 3B 5A 4F 43 07* (Titel des Fensters auf "ZOC" setzen).

Auf den meisten Linux-Systemen könntest du die Sequenz aus dem obigen Beispiel über `echo -e "\e]0;ZOC\a"` ausgeben. In Programmiersprachen, die C-ähnliche Zeichenketten verwenden (C, C++, Perl, PHP, usw.), könntest du etwas Ähnliches verwenden wie `print "\033]0;ZOC\007";`.

Fenstertitel ändern: `ESC ] 0 ; <Titel> ^G`

Die Sequenz ermöglicht es dem entfernten Programm, den Titel des Fensters (oder Tabs), in dem die Sitzung läuft, zu ändern.

Fenstertitel ändern (alternativ): `ESC [ | $ t = <title> CR`

Die Sequenz ist eine alternative Version der oben genannten und ermöglicht es ebenfalls dem entfernten Programm, den Fenstertitel zu ändern.

Bildschirm löschen und Emulation zurücksetzen: `ESC ] R`

Eine Byte-Folge von *hex 1B 5D 52* wird einen Bildschirmlöschvorgang und einen Emulationsreset durchführen (identisch mit der Auswahl von "Bildschirm löschen" im Ansicht-Menü).

Fenstertitel ändern: `ESC ] 52 ; c ; <base64-data> ^G`

Die OSC52-Sequenz ermöglicht es dem entfernten Programm, Text gemäß dem OSC-52-Protokoll in die Zwischenablage zu setzen (erfordert die Aktivierung von 'Lokalen Befehl starten', siehe unten).

Lokalen Befehl starten: `ESC [ | <Befehl> CR`

Ein Host kann diese Sequenz senden, um einen Befehl lokal über ZOC auszuführen.

Lokalen Befehl starten: `ESC [ | <Befehl> CR`

Ein Host kann diese Sequenz senden, um einen Befehl lokal über ZOC auszuführen.

Wichtig: Da dies ein potenzielles Sicherheitsrisiko darstellt, wird die Sequenz ignoriert, es sei denn, die Option zur Aktivierung ist unter [Optionen→Verbindungsprofil→Erweitert](#) gesetzt. Alternativ wird die Sequenz auch verarbeitet, wenn eine Datei mit dem Namen `remoterun.flg` im ZOC-Programmordner (Windows) existiert oder wenn eine Datei mit dem Namen `.zoc9_remoterunflag` im Benutzer-Heimatordner (macOS) existiert.

Der Befehl wird über die CreateProcess-API an Windows übergeben und kann optionale Parameter enthalten, z.B. `notepad.exe newfile.txt` oder über `cmd.exe`

`C:\Windows\system32\cmd.exe /c dir >out.txt`. Unter macOS können Programme direkt ausgeführt werden (`/bin/touch /tmp/didit.txt`). Aufgaben, die Shell-Funktionen erfordern (z.B.

Ausgabeumleitung), müssen über eine Shell ausgeführt werden, z.B. `/bin/bash -c ls ~/Downloads >/tmp/out.dir` oder über eine Shell als Shell-Skript `/bin/bash /Users/you/thetask.sh`

Wenn der Befehlsparameter mit `http://`, `https://` oder `ftp://` beginnt, wird die URL an den Standard-Webbrowser des Benutzers weitergeleitet.

Wenn der Befehl mit dem Text `%SHELLOPEN%` gefolgt von einem Leerzeichen und einem Dateinamen beginnt, wird die Datei über das Betriebssystem mit dem vom Benutzer gewählten Standardprogramm geöffnet, z.B. `%SHELLOPEN% C:\ZOC\piechart.jpg`.

Es gibt auch einige Platzhalter: `DOWNLOADDIR`, `LASTDOWNLOADEDFILE`, `LASTSPOOLFILE`.

Wenn diese zwischen Prozentzeichen erscheinen (z.B. `%LASTDOWNLOADEDFILE%`) oder in Klammern nach einem Dollarzeichen (z.B. `$(LASTSPOOLFILE)`), werden sie durch ihre jeweiligen Werte ersetzt, bevor der Befehl ausgeführt wird. Ein typisches Beispiel könnte sein, dass ein Host einen Bericht an den Benutzer über Zmodem sendet und ihn dann über `%SHELLOPEN% " %DOWNLOADDIR%\thereport.pdf" oder %SHELLOPEN% "%LASTDOWNLOADEDFILE%"` öffnet.

NetTerm-Kompatibilität: `ESC [ ] <parameter> ESC [ <n> *`

ZOC unterstützt eine Reihe von Sequenzen von NetTerm, um Operationen auf dem lokalen Computer, auf dem das Terminal ausgeführt wird, durchzuführen.

Der Parameter `<n>` ist eine Dezimalzahl, die die auszuführende Funktion bestimmt. Die Bedeutung von `<parameter>` hängt also von `<n>` ab, wie unten beschrieben:

0 - URL, die im lokalen Browser geöffnet werden soll, z.B. `ESC [ ] https://www.emtec.com/ ESC [ 0 *`.

1 - Befehl, der lokal ausgeführt werden soll (gleich wie die Sequenz 'Lokalen Befehl starten' oben).

2 - Führe eine ZOC-Aktion aus (dies ist eine Aktionsdefinition, ähnlich wie Aktionen, die du finden wirst, wenn du eine ZOC-Verbindungsprofil-Datei mit einem Editor öffnest, und zum Abschnitt der Benutzertasten gehst).

6 - Verzeichnis, das als Download-Verzeichnis verwendet werden soll.

7 - Verzeichnis, das als Upload-Verzeichnis verwendet werden soll.

8 - Name und Pfad der Datei, die beim nächsten Upload übertragen werden soll.

11 - Text, der in die lokale Zwischenablage geschrieben werden soll.

12 - Vollständiger Pfad und Name einer Datei, die mit dem zugehörigen Standardprogramm für den Dateityp der Datei geöffnet werden soll. Der Platzhalter `%LASTDOWNLOADEDFILE%` kann verwendet werden, um auf die zuletzt empfangene Datei zu verweisen.

17 - Der Text `NOWAIT`. In diesem Fall wird die zuletzt heruntergeladene Datei auf dem lokalen Computer geöffnet (mit dem Standardprogramm für den Dateityp der Datei).

17 - Der Text `PRINT`. Die zuletzt heruntergeladene Datei wird an den im [Programmeinstellungen→Drucker](#) konfigurierten Drucker gesendet.

17 - Der Text `DELETE`. Die zuletzt heruntergeladene Datei wird gelöscht.

Beispiel: `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` wird anschließend (nur für diese Sitzung) heruntergeladene Dateien im Ordner `C:\DOWNLOADS\` speichern.

Beispiel: `ESC [ ] C:\DOWNLOADS\REPORT.PDF ESC [ 12 *` wird die Datei `REPORT.PDF` auf dem lokalen Computer mit der Standardverknüpfung für PDF-Dateien öffnen.

Beispiel: `ESC [ ] ^RUN=postprocess.zrx ESC [ 2 *` wird das REXX-Skript `postprocess.zrx` ausführen.

Beispiel: `ESC [ ] NOWAIT ESC [ 12 *` wird die zuletzt heruntergeladene Datei (z.B. `MYREPORT.PDF`) auf dem lokalen Computer mit der Standardverknüpfung für PDF-Dateien öffnen.

Wichtig: Diese Sequenzen werden nur verarbeitet, wenn `remoterun.flg` auf dem System vorhanden ist oder wenn sie unter [Optionen→Verbindungsprofile→Erweitert](#) aktiviert sind (siehe 'Lokalen Befehl starten' oben). Wenn du deine Verwendung dieser Sequenzen debuggen möchtest, kannst du den Hex-Modus oder ASCII-Trace unter [Optionen→Verbindungsprofile→Trace/Debug](#) aktivieren.

1.11.3 Tastaturbelegung

- ☐ [Tastenbelegung für Menü-Funktionen](#)
- ☐ [Standard-Tastenbelegung für die xterm/VT220/VT420/VT102/Linux Emulationen](#)
- ☐ [Standard-Tastenbelegung für die TN3270 Emulation](#)
- ☐ [Standard-Tastenbelegung für die Wyse/TVI Emulationen](#)
- ☐ [Standard-Tastenbelegung für die TN5250 Emulation](#)
- ☐ [Standard-Tastenbelegung für die Tandem 6530 Emulation](#)
- ☐ [Namen der Tasten für benutzerdefinierte Tastenbelegung](#)
- ☐ [Tastaturprofile, Benutzerdefinierte Tastenbelegung](#)
- ☐ [Senden von Steuerzeichen](#)

Menü-Funktionen

Taste	Funktion
Alt+B	Anzeige → Datenprotokollfenster anzeigen
Alt+C	Eingabezeile ein/aus
Alt+D	Datei → Adressbuch
Alt+E	Editor
Alt+F	Anzeige → Im Zurückblättern suchen
Alt+H	Datei → Verbindung trennen
Alt+K	Anzeige → Zurückblättern löschen
Alt+L	Log in Datei aktiv/inaktiv
Shift+Alt+L	Name der Logdatei festlegen
Alt+O	Datei → Neue Verbindung
Alt+P	Log auf Drucker
Shift+Alt+R	Datei → Bildschirm Drucken
Alt+R	Datei → Wieder verbinden
Shift+Alt+T	Datei → Neues Registerblatt
Shift+Alt+W	Datei → Registerblatt oder Fenster schließen
Alt+X	Programm beenden
Alt+W	Fenster oder Register schließen
Alt+Y	Bildschirm löschen
Strg+Tab	Anzeige → Nächstes Registerblatt
Strg+Shift+Tab	Anzeige → Vorheriges Registerblatt
Shift+Esc	Anzeige → Seitenübersicht
Strg+PgUp	Upload
Strg+Shift+PgUp	ASCII Upload
Strg+PgDn	Download
Shift+BildAuf	Anzeige → Zurückscrollen
Alt+BildAuf	Anzeige → Zurückscrollen im Fenster
Shift+Einf	Bearbeiten → Einfügen
Alt+Einf	Bearbeiten → Einfügen ohne Zeilenwechsel
Alt .	Optionen → Verbindungsprofil bearbeiten
Alt ,	Optionen → Programmeinstellungen
Alt =	Anzeige → Chat Modus

Alt +	REXX Skript starten
Alt -	REXX Skript abbrechen
Alt *	REXX Skript bearbeiten
Alt End	Datei → Break senden

Senden von Control Codes

Tastenkombination	Steuerzeichen
Ctrl+Letter (a-z)	Sendet ^A ... ^Z (hex 01 ... hex 1A)
Ctrl+Shift+Letter (A-Z)	Sendet ^A ... ^Z (hex 01 ... hex 1A)
Ctrl+@ oder Ctrl+Space	Sendet ^@ (hex 00)
Ctrl+[oder Ctrl+3	Sendet ^[(hex 1B)
Ctrl+\ oder Ctrl+4	Sendet ^\ (hex 1C)
Ctrl+] oder Ctrl+5	Sendet ^] (hex 1D)
Ctrl+^ oder Ctrl+~ or Ctrl+6	Sendet ^^ (hex 1E)
Ctrl+[oder Ctrl+? oder Ctrl+7	Sendet ^_ (hex 1F)
Ctrl+8	Sendet DEL (hex 7F)

1.11.4 Xterm/VT220/Linux Sondertasten auf der PC Tastatur

Die folgende Tastenliste zeigt, wie die Tasten der Emulation von ZOC auf die lokale Tastatur abgebildet werden. Dies geschieht automatisch, wenn Sie die Emulation für eine Verbindung wählen. Jedoch kann jede dieser Tasten auf jede andere Tasten-Kombination der lokalen Tastatur zugeordnet werden, indem Sie die Funktion [Optionen→Tastaturprofile](#) verwenden und dabei die [Bezeichnungen für Spezialtasten](#) nutzen. Sie können auch über [Optionen→Verbindungsprofil→Benutzerknöpfe](#) auf benutzerdefinierte anklickbare Bildschirmknöpfe zugeordnet werden.

Standardmäßige Zuordnung der VT/XTerm Spezialtasten

VT220 Taste

[F1] ... [F4]
[PF1] ... [PF4]
[F5] ... [F10]
[F11]/[F12]

[F13] ... [F20]
[Insert Here]

[Remove]
[Select]
[Find]
[Prev Screen]
[Next Screen]

PC Taste

nicht verfügbar sup class="helpsup"[1]/sup
[F1] ... [F4] sup class="helpsup"[1]/sup
[F5] ... [F10] sup class="helpsup"[1]/sup
[F11]/[F12] oder
Shift+[F1]/[F2] (*)
Shift+[F3] ... [F10] sup class="helpsup"[1]/sup
[Einf] (nur Windows)
Shift+[Pos1]
[Entf]
[Pos1]
[Ende]
[BildAuf]
[BildAb]

Tasten am Nummernblock

VT102 Tastatur

[PF1]

PC Nummernblock (NumLock aus)

[F1]

[PF2]
[PF3]
[PF4]
[0] ... [9]
[Return]
[.]
[-]
[.]

[F2]
[F3]
[F4]
[0] ... [9]
[Return]
[./Entf]
[-]
[+]

sup class="helpsup"[1]/sup Falls Sie Probleme mit den standardmäßigen Belegungen der F-Tasten haben, können Sie auch per Tastaturprofil (Optionen Menü) die Tasten mit Werten aus der Tabelle auf [F-Tasten verschiedener Terminalemulatoren](#) belegen.

1.11.5 TN3270 Spezialtasten auf der lokalen Tastatur

Die folgende Tastenliste zeigt, wie die Tasten der Emulation von ZOC auf die lokale Tastatur abgebildet werden. Dies geschieht automatisch, wenn Sie die Emulation für eine Verbindung wählen. Jedoch kann jede dieser Tasten auf jede andere Tasten-Kombination der lokalen Tastatur zugeordnet werden, indem Sie die Funktion [Optionen→Tastaturprofil](#) verwenden und dabei die [Bezeichnungen für Spezialtasten](#) nutzen. Sie können auch über [Optionen→Verbindungsprofil→Benutzerknöpfe](#) auf benutzerdefinierte anklickbare Bildschirmknöpfe zugeordnet werden.

Standardmäßige Zuordnung der TN3270 Spezialtasten

3270 Taste

[Newline]
[Enter/DatenFrg.]

[Reset]

[Ins]

[Attn]

[PA1] ... [PA3]
[EraseEOF]
[Clear]

[EraseInput]
[PF1] ... [PF12]
[PF13] ... [PF24]
[PF7]/[PF8]
[Left2]
[Right2]

PC Taste

[Return] sup class="helpsup"[1]/sup
Shift+[Return] sup class="helpsup"[1]/sup
Alt+[Return] sup class="helpsup"[1]/sup
[Enter] (rechter Nummernblock)
[rechte Strg-Taste] (nur Windows)
[Esc]
[linke Strg-Taste] (nur Windows)
[Ins] (nur Windows)
Shift+[Backspace]
Shift+[Entf]
Shift+[Pos1]
[Strg+C]
Shift+[Pause] (nur Windows)

Alt+[1] ... [3]
[Ende]
Shift+[Esc]
[Pause] (nur Windows)
[Clear] (nur macOS)
Alt+KEY_DELETE
[F1] ... [F12] nbsp;nbsp;
Shift+[F1] ... [F12] nbsp;nbsp;
[BildAuf] / [BildAb]
Shift+[Links]
Shift+[Rechts]

Spezielle ZOC-Funktionen

Die Zuordnungen unten führen Funktionen aus, die auf einem originalen TN3270-Terminal nicht zu finden sind, die jedoch im Kontext eines Terminal-Emulators besonders nützlich sein können, z.B. bietet ZOC eine Undo-Funktion wie in einem lokalen Editor oder in einem Textverarbeitungsprogramm.

ZOC 3270 Funktion

[HomeCmd]
[MovePrevWord]
[MoveNextWord]
[DeleteNextWord]
[MoveEOF]
[MoveBOL]
[SelectAll]
[EraseEOF]
[EraseField]
[SaveCursorPos]
[RestoreCursorPos]

[Undo]
[Redo]

PC Taste

Alt+[Pos1]
Alt+[Links]
Alt+[Rechts]
Ctrl+Del
Ctrl+[Ende]
Alt+[Enter]
Ctrl+A
Ctrl+K
Ctrl+L
Ctrl+S
Ctrl+R

Ctrl+Z
Ctrl+Shift+Z

sup class="helpsup"[1]/sup Die Funktion der Tasten Return und Shift+Return (auch Alt+Return) kann in den Einstellungen der TN5250 Emulation vertauscht werden. Die Kombination Cmd+[Return] (nur macOS) ist von der Option zum Tauschen der Tasten nicht betroffen.

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 is done by using the defined names for these from [this table](#).

Main Keyboard Keys:

Wyse/Televideo Key-Mappings

[F1] ... [PF12]
[F13] ... [F24]
[InsChar]
[DelChar]
[Transmit]

Equivalent on the PC Keyboard

[F1] ... [F12]
[Shift+F1] ... [Shift+F12]
[Shift+Ins]
[Shift+Del]
[Alt+Enter]

1.11.7 TN5250 Spezialtasten auf der PC Tastatur

Die folgende Tastenliste zeigt, wie die Tasten der Emulation von ZOC auf die lokale Tastatur abgebildet werden. Dies geschieht automatisch, wenn Sie die Emulation für eine Verbindung wählen.

Jedoch kann jede dieser Tasten auf jede andere Tasten-Kombination der lokalen Tastatur zugeordnet werden, indem Sie die Funktion [Optionen→Tastaturprofile](#) verwenden und dabei die [Bezeichnungen für Spezialtasten](#) nutzen. Sie können auch über [Optionen→Verbindungsprofil→Benutzerknöpfe](#) auf benutzerdefinierte anklickbare Bildschirmknöpfe zugeordnet werden.

Standardmäßige Zuordnung der TN5250 Spezialtasten

5250 Taste

PC Taste

[PF1] ... [PF12]	[F1] ... [F12]
[PF13] ... [PF24]	Shift+[F1] ... Shift+[F12]
[Newline]	[Return] (*)
[Enter/DatenFrg.]	Shift+[Return] (*)
	Alt+[Return] (*)
	[rechte Strg-Taste] (nur Windows)
	Cmd+[Return] (*) (nur macOS)
[Field Exit]	[Enter] (rechter Nummernblock)
[Reset]	[Esc] oder [linke Strg-Taste]
[Ins]	[Ins] (nur Windows)
	Shift+[Backspace]
	Shift+[Entf]
	Shift+[Pos1]
[EraseEOF]	Alt+[Ende]

(*) Die Funktion der Tasten Return und Shift+Return (auch Alt+Return) kann in den Einstellungen der TN5250 Emulation vertauscht werden. Die Kombination Cmd+[Return] (nur macOS) ist von der Option zum Tauschen der Tasten nicht betroffen.

5250 Tasten via Esc-Kombination

In einigen Terminalemulationen können Spezialtasten der 5250 auch durch eine Zwei-Tasten Sequenz mit der Esc-Taste simuliert werden.

Dies ist auch mit ZOC möglich, jedoch nutzt ZOC die Esc Taste standardmäßig für Reset. Jedoch kann in [Optionen→Tastaturprofile](#) die Esc-Taste alternativ mit [EscapeMenu](#) belegt werden. Mit dieser Einstellungen ist es dann möglich, nach Esc eine der unten aufgeführten Tasten zu drücken um die 5250 Sondertasten zu simulieren (so kann dann z.B. [Attention](#) durch Drücken von [Esc A](#) übergeben werden). Allerdings ist es wahrscheinlich einfacher, die entsprechenden Tasten (z.B. [Attention](#)) unter [Optionen→Tastaturprofile](#), mit Hilfe der [Bezeichnungen für Spezialtasten](#) direkt auf eine geeignete PC-Taste zu legen.

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 Sondertasten auf der PC-Tastatur

Die folgende Tastenliste zeigt, wie die Tasten der Emulation von ZOC auf die lokale Tastatur abgebildet werden. Dies geschieht automatisch, wenn Sie die Emulation für eine Verbindung wählen. Jedoch kann jede dieser Tasten auf jede andere Tasten-Kombination der lokalen Tastatur zugeordnet werden, indem Sie die Funktion [Optionen→Tastaturprofile](#) verwenden und dabei die [Bezeichnungen für Spezialtasten](#) nutzen. Sie können auch über [Optionen→Verbindungsprofil→Benutzerknöpfe](#) auf benutzerdefinierte anklickbare Bildschirmknöpfe zugeordnet werden.

Standardmäßige Zuordnung der Tandem Spezialtasten

Tandem 6530 Taste

[F1] ... [F12]
[F13] ... [F16]
[Enter]

PC Taste

[F1] ... [F12]
[Alt+F3] ... [Alt+F6]
[Return]
Shift+[Return]
Ctrl+[Return]

1.11.9 Bezeichnungen für Emulationstasten

Alle Emulationen innerhalb von ZOC verfügen über voreingestellte Tastenzuordnungen, bei denen sie Emulations-spezifische Tasten (z.B. die VT220 Next-Taste) automatisch auf eine entsprechende (oder geeignete) PC- oder Mac-Taste abbilden (siehe [Tastatur Default Zuordnungen](#)).

Alle Tasten dieser Emulationen können auch in einem anderen Kontext verwendet werden, z.B. bei benutzerdefinierten Tastenzuordnungen in [Optionen→Tastaturprofilen](#), für die REXX-Funktion [ZocSendEmulationKey](#), oder für [Optionen→Verbindungsprofil→Benutzerknöpfe](#). Dies geschieht über festgelegte Bezeichnungen der Tasten (s.u.), die verfügbar sind, während die entsprechende Emulation aktiviert ist.

Namen der Sondertasten in Emulationen

Emulation

VT220/VT420/VT520/VT102

Verfügbare Tastenbezeichnungen

Esc, PF1, PF2, PF3, PF4, F5, F6, ... F20, Help (= F15), Do (= 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.

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.

TN3270

Insert, Enter, Tab, Backspace, BackDel, BackTab, Reset, Home, Left, Left2, Right, Right2, Up, Down, NewLine, Attn, Pa1, Pa2, Pa3, Clear, Delete, EraseEOF, MoveEOF, MonoCase, Dup, FieldMark, SysReq, EraseInput, CursorSelect, PF1, PF2, ..., PF24.
ZOC Erweiterungen: Undo, Redo, HomeCmd, LowerHome, HomeAny, MoveNextWord, MovePrevWord, DeleteNextWord, MoveBOL, EraseField, Lightpen, CharNot ($\neg$), MoveTo:r/c (r/c= row/column).

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.

Linux

Esc, Up, Down, Right, Left, Home, End, Insert, ModUp, ModDown, ModRight, ModLeft, Enter, Return, Backspace, Del, Delete, PgUp (oder Prev), PgDown (oder Next), Tab, BackTab, F1, F2, ..., F20.

Sun CDE

Wie VT220. Allerdings muss KF1-KF4 statt PF1-PF4 benutzt werden.

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.

Beispiel: Wenn die VT220 Emulation aktiv ist, können Sie Ctrl+I mit der Insert-Funktion belegen indem Sie im Optionen-Menü die Funktion Tastaturprofile wählen. Dort klicken Sie auf Strg+I, dann auf den Knopf Weitere Funktionen und geben `Insert` mit dem Typ `Name einer Emulationstaste` an.

Von einem REXX-Script aus können Sie in der Skriptverarbeitung die Insert Funktion durch den Befehl `Call ZocSendEmulationKey "Insert"` auslösen.

Falls Sie mit den vorgegebenen Codes und Namen Probleme haben, können Sie in den Tastaturprofilen (Optionen-Menü) für die Tasten auch spezifische Codes hinterlegen und z.B. die F-Tasten mit den Codes aus der folgenden Tabelle belegen: [F-Tasten verschiedener Terminalemulatoren](#).

1.11.10 ZOC REXX-Befehle/Funktionen (alphabetisch)

Dieser Abschnitt beschreibt in alphabetischer Reihenfolge die ZOC-Erweiterungen zur REXX-Sprache. Alternativ gibt es auch eine [Liste von ZOC-REXX Befehle nach Kategorien](#)

REXX ist ansich eine vollständige Programmiersprache mit Variablen, Schleifen usw. (siehe [REXX-Sprachelemente](#)). Dieser Grundumfang wird von ZOC durch die untenstehenden Befehle für die Terminalemulation erweitert.

Der Übersichtlichkeit halber werden in den unten folgenden Beispielen die REXX Sprachelemente in Großschrift verwendet, z.B. `CALL`, `IF`, `SAY`. ZOC-Erweiterungen werden in `CamelCase` angegeben, z.B. `ZocAsk`, `ZocWrite`. Namen die im Programm vom Benutzer selbst vergeben werden (z.B. Variablen, Prozeduren, etc.), stehen in Kleinbuchstaben (z.B. `antwort= ZocAsk("Ihr Name?")`).

Aufruf von ZOC-Erweiterungen

Die Namen der ZOC-Erweiterungen beginnen immer mit `Zoc`. Sie gleichen internen Prozeduren/Funktionen und für die meisten davon müssen ein oder mehrere Argumente angegeben werden.

Grundsätzlich gibt es zwei Arten von ZOC-Befehle: Befehle, die einen Wert zurückgeben und Befehle, die keinen Wert zurückgeben.

- ❑ ZOC-Befehle, die keinen Wert zurückliefern, werden wie Prozeduren aufgerufen:
`CALL <befehlsname> <argumente>`, z.B. `CALL ZocDelay 2.5`.
- ❑ Funktionen werden anhand der Funktions-Syntax aufgerufen:
`<ergebnis-var>= <befehlsname>(<argumente>)`, z.B. `antwort= ZocAsk("Ihr Name?")`

Allerdings ist es möglich, Funktionen (also Befehle mit Rückgabewert) wie Prozeduren aufzurufen, wenn der Rückgabewert nicht von Interesse ist. Daher sind z.B. sowohl die Aufrufe `CALL ZocDownload "ZMODEM", "DATENORDNER"` als auch `error= ZocDownload("ZMODEM", "DATENORDNER")` erlaubt. Zusätzlich erlaubt die ZOC-REXX Implementierung sogar die folgende Variante: `CALL ZocDownload("ZMODEM", "SOMEFOLDER")`

ZOC REXX Erweiterungen

In der nachfolgenden alphabetisch sortierten Befehlsliste sind diejenigen ZOC-Erweiterungen, die signifikante Rückgabewerte haben, dadurch gekennzeichnet, dass sie in den Syntax-Beispielen in Funktionsform (mit Klammern) geschrieben sind.

`ZocAsk([<titel> [, <vorgabe>]])`

Öffnet ein Texteingabefenster und liest Text vom Benutzer ein.

Bei Angabe eines Arguments `<vorgabe>` wird das Eingabefeld mit diesem Text vorbelegt.

Beispiel:

```
answer= ZocAsk("Das beste Terminal?", "ZOC")
IF answer= "ZOC" THEN ...
```

Siehe auch: [ZocDialog](#), [ZocAskPassword](#), [ZocAskFilename](#), [ZocAskFoldername](#), [ZocRequest](#), [ZocRequestList](#)

`ZocAskPassword([<titel>])`

Ähnlich dem Befehl [ZocAsk](#), dient jedoch zur Kennworteingabe, d.h. eingegebenen Zeichen werden nur als Platzhalter angezeigt. Außerdem kann das Feld nicht vorbelegt werden.

Beispiel:

```
pw= ZocAskPassword("Ihr Passwort?")
IF pw=="geheim" THEN ...
```

Siehe auch: [ZocAsk](#), [ZocDialog](#), [ZocAskFilename](#), [ZocRequest](#), [ZocRequestList](#)

ZocAskFilename(<titel> [, <vorgabe>])

Öffnet einen Dateiauswahldialog und liefert den ausgewählten Dateinamen zurück.

Bei Abbruch des Dialogs wird der Text ##CANCEL## zurückgeliefert.

Beispiel:

```
file= ZocAskFilename("Welche Datei übertragen?", "*.ZIP")
IF file\="##CANCEL##" THEN DO
    CALL ZocUpload "ZMODEM", file
END
```

Siehe auch: [ZocAsk](#), [ZocDialog](#), [ZocFilename](#), [ZocAskFileNames](#), [ZocAskFoldername](#), [ZocListFiles](#), [ZocMessageBox](#), [ZocRequest](#), [ZocRequestList](#)

ZocAskFileNames(<title> [, <vorgabe> [, <trennzeichen>]])

Öffnet einen Dateidialog und liefert die Liste der selektierten Dateinamen zurück. Die <vorgabe> kann einen Ordner oder eine vorselektierte Datei angeben. Bei Abbruch des Dialogs wird der Text ##CANCEL## zurückgeliefert.

Die Namen können per REXX Funktion `ZocString("WORD", idx)` aus der Liste extrahiert werden. Falls Sie Dateinamen erwarten, die Leerzeichen im Namen haben, muss ein anderes Trennzeichen angegeben werden. Die Namen können dann stattdessen mittels der Funktion `ZocString("PART", ...)` aus der Liste geholt werden.

Beispiel:

```
files= ZocAskFileNames("Welche Dateien bearbeiten?", "*.ZIP", "|")

anzahl= ZocString("PARTCOUNT", files, "|")
DO i=1 TO anzahl
    name= ZocString("PART", files, i, "|")
    SAY i||". NAME= "||name
END
```

Siehe auch: [ZocAsk](#), [ZocFilename](#), [ZocAskFilename](#), [ZocAskFoldername](#), [ZocListFiles](#), [ZocMessageBox](#), [ZocRequest](#), [ZocRequestList](#)

ZocAskFoldername(<titel> [, <vorgabe>])

Öffnet einen Dialog zur Auswahl eines Ordners und liefert den gewählten Namen zurück. Die optionale <vorgabe> ist ein Ordner, der beim Anzeigen des Dialogs vorselektiert ist. Bei Abbruch des Dialogs wird der Text ##CANCEL## zurückgeliefert.

Beispiel:

```
folder= ZocAskFoldername("Select Folder")
IF folder\="##CANCEL##" THEN DO
    SAY folder
```

END

Siehe auch: [ZocFilename](#), [ZocAskFilename](#), [ZocAskFileNames](#), [ZocListFiles](#)

ZocBeep <n>

Erzeugt n Signaltöne.

Beispiel:

```
CALL ZocBeep 2
```

ZocClipboard <kommando> [, <zeichenkette>]

Führt eine Funktion der folgenden Funktion der Zwischenablage aus.

READ

Liefert den Inhalt der Zwischenablage

WRITE

Schreibt den Wert des zweiten Parameters in die Zwischenablage

Beispiel:

```
clip= ZocClipboard("READ")
newclip= clip||ZocCtrlString("^M^J Zoc war hier!")
CALL ZocClipboard "WRITE", newclip
```

ZocClearScreen

Löscht den Bildschirm und setzt die momentan aktive Emulation in den Ursprungszustand zurück.

Beispiel:

```
CALL ZocClearScreen
```

ZocCommand <funktion>

Führt eine ZOC-interne Funktion aus. Gültige Werte für <funktion> sind:

CLS

Bildschirm löschen.

CLEARSCROLLBACK

Inhalt des Scroll-Back Puffers löschen.

CANCELCONNECT

Momentan laufenden Verbindungsaufbau abbrechen.

DISCONNECT

Aktuelle Verbindung unterbrechen.

LOADGLOBALCOLORS

Lädt die globale Farbtabelle aus einer Datei.

SAVEPROGRAMSETTINGS

Speichert mit [ZocSetProgramOption](#) gemachte Änderungen permanent ab.

SAVESESSIONPROFILE

Speichert Änderungen am Verbindungsprofil (durch [ZocSetSessionOption](#)) in die Datei.

SEENDBREAK

Sendet ein Break-Signal an das Modem (nur mit Verbindungstyp Serial/Modem).

SETMARKEDAREA

Markiert einen Bildschirmbereich. Nach dem Funktionsnamen folgt entweder das Wort **BLOCK** oder **STREAM** und danach die Koordinaten x1,y1 und x2,y2.

Beispiel:

```
CALL ZocCommand "SAVESESSIONPROFILE"
```

Beispiel:

```
CALL ZocCommand "SETMARKEDAREA", "BLOCK", 0,0, 1,79
```

Siehe auch: [ZocMenuEvent](#)

ZocConnect [<gegenstelle>]

Baut eine Verbindung zur angegebenen Gegenstelle auf. Ist der Parameter nicht angegeben, erscheint ein Eingabefenster. Als Verbindungstyp dient entweder die aktuell im Verbindungsprofil eingestellte Verbindungstyp, bzw. kann dieser mittels des Befehls [ZocSetDevice](#) festgelegt werden.

Hinweis: Bei SSH-Verbindungen kann die Login-Information in der folgenden Form mitgegeben werden: `CALL ZocConnect "user:pass@ssh.somedomain.com"`. Falls nötig können Sie auch den Namen einer Schlüsseldatei mit Doppelpunkt getrennt hinter dem Kennwort angeben: `CALL ZocConnect "user:pass:id_rsa@ssh.somedomain.com"`.

Beispiel:

```
CALL ZocSetDevice "Secure Shell"  
CALL ZocConnect "harry:alohomora@192.168.1.1:10022"
```

Beispiel:

```
CALL ZocSetDevice "Telnet"  
CALL ZocConnect "server.hogwarts.edu"  
  
Call ZocTimeout 20  
x= ZocWait("Login:")
```

```

IF x=640 THEN SIGNAL waitfailed /* login-prompt kam nicht */
CALL ZocSend "harry^M"
x= ZocWait("Password:")
IF x=640 THEN SIGNAL waitfailed /* password-prompt kam nicht */
CALL ZocSend "alohomora^M"

/* here we are logged in */

waitfailed:
EXIT

```

Siehe auch: [ZocConnectHostdirEntry](#), [ZocSetDevice](#), [ZocDisconnect](#), [ZocGetInfo\("ONLINE"\)](#)

ZocConnectHostdirEntry <name>

Stellt eine Verbindung über einen Eintrag im Adressbuch an (der Eintrag sollte nicht mit einer AutoLogin REXX-Datei verknüpft sein).

Beispiel:

```
CALL ZocConnectHostdirEntry "ZOC-Support-Mailbox"
```

Siehe auch: [ZocConnect](#), [ZocDisconnect](#), [ZocGetInfo\("ONLINE"\)](#)

ZocCtrlString(<text>)

Diese Funktion konvertiert eine Zeichenkette die Control-Codes enthält in eine Zeichenkette, bei der die [Steuerzeichen](#) durch ihre eigentlichen Werte/Bytes ersetzt sind.

Beispiel:

```
crlf= ZocCtrlString("^M^J") /* crlf wird hex"0D0A" */
```

Siehe auch: [ZocString](#)

ZocDdeClient([<kanal>], <unterkommando> [, <parameter>])

Diese Funktion kann dazu verwendet werden, um mit anderer Software über DDE (Dynamic Data Exchange) zu interagieren. MS-Excel unterstützt z.B. DDE und der DDE-Client kann dazu verwendet werden, um zum Beispiel Daten aus einem Excel Arbeitsblatt abzufragen.

Mögliche [<unterkommandos>](#) sind:

INIT

Initialisiert eine DDE Verbindung. INIT folgen zwei Parameter: "DDE-Server" und "Topic". INIT liefert entweder ##ERROR## zurück, oder eine Kanalnummer für nachfolgende Befehle.

EXECUTE

Führt eine Serverfunktion aus. Auf EXECUTE folgt ein Parameter: "Execution-Command" (z.B. ein Excel Befehl).

REQUEST

Ruft Daten vom Server ab. REQUEST folgt ein Parameter: "Data-Address" (z.B. eine Excel-

Zelle oder ein Bereich).

CLOSE

Schließt einen geöffneten DDE-Kanal.

Beispiel:

```
chan= ZocDdeClient("INIT", "EXCEL", "Tabelle1")
SAY "INIT: channel= "||chan

IF chan\="##ERROR##" THEN DO
  data= ZocDdeClient(chan, "REQUEST", "Z1S1")
  SAY "REQUEST-DATA: "||data

  CALL ZocDdeClient chan, "CLOSE"
END
EXIT
```

ZocDelay [<dauer>]

Wartet die angegebene <dauer> in Sekunden oder 200 Millisekunden (falls kein Parameter angegeben wurde).

Bruchteile von Sekunden können mit Dezimalpunkt (nicht Komma) dargestellt werden.

Beispiel:

```
CALL ZocDelay 4.8
```

ZocDeviceControl <string>

Dieser Befehl gibt ein Kommando an eines der ZOC Kommunikationsgeräte um dort spezielle Funktionen auszuführen (z.B. Abfrage der Signale am COM-Port bei Verbindungstyp Serial/Modem).

Mögliche Funktionen der einzelnen Kommunikationsmethoden sind unter [Verbindungsmethoden](#) beschrieben.

Beispiel:

```
state= ZocDeviceControl("GETRS232SIGNALS")
```

Siehe auch: [ZocSetDevice](#)

ZocDialog <subcommand> [, <parameter>]

Führt eine Dialog-Funktion aus:

LOAD

Lädt ein benutzerdefiniertes Dialogfenster. Der zweite Parameter ist eine Kombination aus Dialognamen und dem Dateinamen der Dialog-Vorlage-Datei (der Dateiname muss entweder den kompletten Dateipfad enthalten, oder die Datei muss sich im Verzeichnis befinden, in dem

sich das laufende Script befindet).

SHOW

Zeigt ein benutzerdefiniertes (vorher geladenes) Dialogfenster an. Optional kann ein zweiter Parameter wie bei [LOAD](#) angegeben werden, in dem Fall beinhaltet der [SHOW](#) Befehl das Laden, d.h. es muss vorher kein gesonderter [LOAD](#) Befehl verwendet werden.

GET

Liefert den Wert eines Dialog-Elements zurück (z.B. den Text, der vom Benutzer in ein Eingabefeld geschrieben wurde, oder den Zustand einer Checkbox oder eines Radiobuttons).

SET

Setzt den Startwert eines Dialog-Elements (z.B. den Text, der dem Benutzer in einem Eingabefeld angezeigt wird oder die Beschriftung. Bei einer Checkbox oder einem Radiobutton kann mit den Werten `"##ON##"` oder `"##OFF##"` auch der Anfangszustand (aktiviert/deaktiviert) gesetzt werden).

Detaillierte Informationen zur Erstellung von Dialog-Vorlagen finden Sie unter [ZocDialog-Vorlagen](#).

Beispiel:

```
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
```

Beispiel:

```
dlgrc= ZocDialog("LOAD", "MAIN@test.dlg")
SAY "Dialog load result: "||dlgrc
CALL ZocDialog "SET", "ED1", "Test"
CALL ZocDialog "SET", "CB1", "1"
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; ergebnisse bearbeiten &ast;/
END
```

Siehe auch: [ZocAsk](#), [ZocAskPassword](#), [ZocMessageBox](#), [ZocRequest](#), [ZocRequestList](#)

ZocDisconnect

Beendet die momentane Verbindung. Identisch mit `ZocCommand "DISCONNECT"`.

Beispiel:

```
CALL ZocDisconnect
```

Siehe auch: [ZocConnect](#), [ZocConnectHostdirEntry](#)

ZocDownload(<protokoll>[:<optionen>], <datei oder verz>)

Datei-Download einer oder mehrerer Dateien unter Verwendung eines Protokolls. Der erste Parameter ist der Name des Übertragungs-Protokolls wie in der Liste bei [Optionen→Verbindungsprofil→Dateitransfer](#).

Der zweite Parameter kann einen Dateinamen oder Pfad zum Speichern der empfangenen Dateien enthalten (siehe Hinweis unten).

Der Parameter <optionen> ist beim Befehl [ZocUpload](#) ausführlich erklärt.

Diese Funktion liefert je nach Erfolg oder Fehler die Zeichenkette ##OK## oder ##ERROR## zurück.

Beispiel:

```
CALL ZocSetSessionOption "TransferAutoStart=no"
ret= ZocDownload("ZMODEM", "C:\ZOC\INFILES")
IF ret=="##ERROR##" THEN DO
    CALL ZocBeep 5
    SAY "Download fehlgeschlagen."
END
```

Hinweis: Der zweite Parameter hängt von der Art des Transfers ab:

XMODEM: Lokaler Dateiname (Zieldatei).

YMODEM: Lokaler Ordner (Zielordner).

ZMODEM: Lokaler Ordner (Zielordner).

SCP: Entfernter Dateiname (Quelldatei), z.B. /var/log/somefile.txt

Hinweis: Wenn in den Transfer-Optionen die Funktion Autostart eingeschaltet ist und die Gegenstelle den Transfer startet, noch bevor ZOC-REXX den ZocDownload-Befehl verarbeitet, öffnen sich zwei Download-Fenster. Es muss deshalb sichergestellt sein, dass ZocDownload ausgeführt wird bevor die Gegenstelle anfängt zu senden, oder dass die Autostart-Option ausgeschaltet ist.

Hinweis: Falls die Datei einen Namen hat, der in [Optionen→Verbindungsprofil→Dateibehandlung](#) zur Speicherung im alternativen Download-Verzeichnis vorgesehen ist, wird der Verzeichnis-Parameter ignoriert.

Siehe auch: [ZocUpload](#)

ZocDoString(<aktionscode>)

Übergibt einen Aktionscode zur Ausführung an ZOC.

Sie können eine solche Zeichenkette erstellen, indem Sie unter [Optionen→Tastaturprofil](#) temporär eine Taste mittels des dortigen Aktions-Assistenten belegen, das Profil abspeichern und danach den dort erzeugten Wert aus der Verbindungsprofildatei (z.B. [Standard.zky](#)) kopieren.

Beispiel:

```
CALL ZocDoString "^EXEC=notepad.exe"
```

Siehe auch: [ZocMenuEvent](#), [ZocShell](#), [ZocSendEmulationKey](#)

ZocEventSemaphore(<unterkommando>[, <signal-id>])

Diese Funktion kann verwendet werden, um Signale zwischen gleichzeitig laufenden REXX Skripten auszutauschen, bzw. diese zu synchronisieren. Sie bietet einen Signal-Mechanismus mit insgesamt 16 Signalen, auf die andere Skripte warten können.

Mögliche Werte für <unterkommando> sind:

RESET

Setzt den Signalisierungszustand des eigenen Skripts zurück.

FIRE

Schickt ein Signal an alle Skripte.

TEST

Liefert die Anzahl der Signale, seit dem letzten Reset.

WAIT

Wartet auf ein Signal.

Wenn seit dem letzten Reset bereits mind. ein Signal eingetroffen ist, kommt die Funktion sofort zurück und liefert den Signalzähler (max 255) und setzt das Signal zurück.

Ansonsten wartet die Funktion, bis entweder ein Signal eintrifft, oder bis das Timeout abläuft (siehe [ZocTimeout](#), in diesem Fall ist der Returncode 640).

<signal-id>

Eine optionale Nummer [1..15] um ein alternatives Signal zu verwenden (Standard ist 0).

Beispiel:

```
Call ZocEventSemaphore "RESET"
/* do some work */
...
/* Warte auf Signal (FIRE) von anderem Script */
ret= ZocEventSemaphore("WAIT")
IF ret\=640 THEN DO
    /* Signal erhalten */
END
```

Siehe auch: [ZocTimeout](#), [ZocWait](#), [ZocGlobalStore](#)

ZocFilename(<befehl>[, <options>])

Diese Gruppe von Funktionen bietet Dateinamensoperationen.

COMBINE <path>[, <path2>], <file>

Kombiniert die Namensteile zu einem Dateinamen. Falls <file> bereits ein voll qualifizierter Dateiname ist, wird <path> und ggf. <path2> ignoriert.

EXISTS, <dateiname>

Liefert ##YES## bzw. ##NO##, je nachdem ob die Datei existiert.

GETFILE, <dateiname>

Liefert nur den Dateinamensteil eines Dateinamens.

GETPATH, <dateiname>

Liefert nur den Verzeichnisteil eines Dateinamens.

GETVOLUMELABEL, <laufwerk>

Liefert die Bezeichnung eines Laufwerks, z.B. "C:" (nur Windows).

GETSIZE, <dateiname>

Liefert die Größe einer Datei.

ISFOLDER <pfadname>

Liefert ##YES## oder ##NO##, je nachdem ob der angegebene Pfad ein Ordner oder eine Datei ist.

RESOLVE, <string>

Löst eine der speziellen ZOC Dateinamensvariablen wie %ZOCFILES% auf (die anderen ZocFilename Funktionen tun dies automatisch).

WRITEACCESS <dateiname>

Liefert ##YES## oder ##NO##, je nachdem ob Schreibrechte für die Datei bestehen.

Beispiel:

```
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(<dateiname>, <ziel>)

Kopiert eine Datei an einen anderen Ort. Ziel kann entweder ein Ordner oder Dateiname sein kann (Wildcards wie * oder ? werden nicht unterstützt, siehe ggf. [ZocListFiles](#)).

Beispiel:

```
CALL ZocFileCopy "Z:\SALES.DAT", "Z:\SALES.BAK"

ok= ZocFileCopy("C:\DATA\USERFILE.TMP", "C:\BACKUP")
IF ok\="##OK##" THEN EXIT
```

Siehe auch: [ZocFilename](#), [ZocFileDelete](#), [ZocFileRename](#), [ZocListFiles](#), [ZocShell](#)

ZocFileDelete(<dateiname>)

Löscht eine Datei. Der Dateiname darf keine Jokerzeichen (* oder ?) enthalten (siehe ggf. [ZocListFiles](#)).

Rückgabewerte sind ##OK## oder ##ERROR##

Beispiel:

```
ok= ZocFileDelete("C:\DATA\USERFILE.TMP")
IF ok\="##OK##" THEN EXIT

filename= ZocFilename("COMBINE", "%ZOCFILES%", "rexx.log")
CALL ZocFileDelete filename
```

Siehe auch: [ZocFilename](#), [ZocFileRename](#), [ZocFileCopy](#), [ZocListFiles](#), [ZocShell](#)

ZocFileRename(<oldname>, <newname>)

Benennt eine Datei um. Die umbenannte Datei befindet sich im selben Verzeichnis wie die ursprüngliche Datei. Dateinamen dürfen keine Jokerzeichen (* oder ?) enthalten (siehe ggf. [ZocListFiles](#)).

Rückgabewerte sind ##OK## oder ##ERROR##

Beispiel:

```
ret= ZocFileRename("C:\DATA\USERFILE.TMP", "C:\DATA\USERFILE.TXT")
```

Siehe auch: [ZocFilename](#), [ZocFileDelete](#), [ZocFileCopy](#), [ZocListFiles](#), [ZocShell](#)

ZocGetHostEntry(<name>, <schlüsselwort>)

Liefert die Zeichenkette für eine Einstellung eines Adressbucheintrags im Schlüsselwort-Wert Format (siehe hierzu auch die Befehle [ZocSetHostEntry](#) oder [ZocSetSessionOption](#)).

Beispiel:

```
pair= ZocGetHostEntry("ZOC-Support-Mailbox", "connectto")
PARSE VALUE pair WITH key="'val' ' '
CALL ZocConnect val
```

ZocGetInfo(<detail>)

Abhängig vom Parameter, liefert diese Funktion eine entsprechende Information über die

Systemumgebung und über den internen Zustand von ZOC.

COMPUTERNAME

Der Name des Rechners auf dem ZOC läuft.

CONNECTEDTO

Der Name des Eintrags im Adressbuch, oder Hostname oder die Telefonnummer mit der wir verbunden sind.

CONNECTEDTORAW

Der Inhalt des "Verbinden mit" Feldes, mit dem die aktuelle Verbindung aufgebaut wurde (z.B. `linux.hogwarts.com`).

CONNECTEDTOIP

Bei IP-basierten Verbindungen, die IP Adresse der Gegenstelle (auch wenn die Verbindung über DNS Namen aufgebaut wurde).

CURRENTDEVICE

The name of the currently active communication method, e.g. `Telnet`.

CURRENTEMULATION

The name of the currently active emulation, e.g. `Xterm`.

CURRENTLOGFILENAME

Aktueller Dateiname für das Log (ohne Pfad).

CURRENTSCRIPTNAME

Dateiname und Pfad des Hauptskripts das momentan läuft (Unterskripte die mit CALL aufgerufen wurden, werden von diesem Befehl nicht zurückgemeldet).

CURRENTSESSIONPROFILENAME

Der Dateiname und Pfad des aktuellen Verbindungsprofils.

CURSOR-X

Die X-Position des Cursors am Bildschirm (beginnend mit 0).

CURSOR-Y

Die Y-Position des Cursors am Bildschirm (beginnend mit 0).

DESKTOPSIZE

Die Nettogröße des Windows/macOS Desktop in Pixel (ohne Taskbar, Dock, etc.)

DOMAINNAME

Der Name der Domäne oder Arbeitsgruppe in der dieser Computer angemeldet ist.

DOWNLOADDIR

Das Laufwerk und Verzeichnis für Downloads.

EXEDIR

Das Verzeichnis, in dem ZOC installiert ist.

LASTDOWNLOADEDFILE

Name und Ordner der letzten Datei die gedownloadet wurde.

MARKEDAREA

Die Start/End Position und Mode der Bildschirmmarkierung in der Form `x1,y1,x2,y2,mode` (Positionen starten bei Null) oder der Text `##NONE##`.

ONLINE

Information, ob momentan eine Verbindung besteht: `##YES##`, `##NO##`, `##UNKNOWN##`.

OSYS

Information über das aktuelle Betriebssystem, z.B. `Windows 10`.

OWNIP

Die IP-V4 Adresse des Computers im lokalen LAN oder WLAN.

OWNIPS

Eine Liste all IP Adressen des lokalen Computers.

PROCESSID

Die Prozess-ID, die das Betriebssystem ZOC zugeteilt hat.

SCREENHEIGHT

Höhe (Anzahl Zeilen) des Terminal-Fensters.

SCREENWIDTH

Breite (Anzahl Zeichen pro Zeile) des Terminal-Fensters.

TRANSFER

Information, ob momentan eine Dateiübertragung läuft: `##YES##`, `##NO##`.

TN3270FIELDATTR x y

Die TN3270 Feldattribute und EA-Farben an einer gegebenen Stelle am Bildschirm (x,y sind nullbasierte Zahlen). Ergebnis in der Form `Attr Foreground Background Flags ....`

UPLOADDIR

Das Laufwerk und Verzeichnis für Uploads.

USERNAME

Der Anmelde-name bzw Userid des aktuell angemeldeten Benutzers.

USERID

(wie USERNAME)

VERSION

Die ZOC-Versionsnummer, z.B. `7.24`

VERSIONEX

Die ZOC-Versionsnummer incl. Beta-Version (falls vorhanden), z.B. `7.24b`

WINPOS

Position und Größe des ZOC-Fensters (in Pixel).

WORKDIR

Das Arbeitsverzeichnis, in dem Optionen usw. gespeichert werden.

Beispiel:

```
ver= ZocGetInfo("VERSIONEX")
SAY "ZOC "||ver

CALL ZocTimeout 30
timeout= ZocWait("Passwort")
IF timeout=640 | ZocGetInfo("ONLINE")<>"##YES##" THEN DO
    SIGNAL PANIC /* Verbindungsabbruch */
END
```

ZocGetProgramOption(<schlüsselwort>)

Liefert die Zeichenkette für eine ZOC-Einstellung im Schlüsselwort-Wert Format (Details zu diesem Format in der Beschreibung des Befehls [ZocGetSessionOption](#) und [ZocSetProgramOption](#)).

Beispiel:

```
pair= ZocGetProgramOption("DisconEndProg")
PARSE VALUE pair WITH key="value
IF value=="yes" THEN DO
    SAY "ZOC wird nach dieser Verbindung beendet."
END
```

Siehe auch: [ZocGetProgramOption](#), [ZocGetSessionOption](#), [ZocSetProgramOption](#)

ZocGetScreen(<x>,<y>,<länge>)

Diese Funktion kann verwendet werden, um Zeichen auszulesen, die aktuell auf dem ZOC-Bildschirm angezeigt werden. Die Funktion liefert <länge> Zeichen ab der Bildschirmposition <x>,<y> (0,0 = links/oben) zurück und springt ohne Anhängen von CR oder LF automatisch in nächste Zeilen. Alternativ gibt es einige Abkürzungen für häufige Kombinationen von x,y und Länge:

ALL: Der gesamte Bildschirm (d.h. von Position 0/0 in der Länge

SCREENWIDTH*SCREENHEIGHT)

LEFTTOFCURSOR: Der Text links vom Cursor (Position= 0/CURSOR-Y, Länge= CURSOR-X)

CURRENTLINE: Die ganze Zeile in der der Cursor steht (Position= 0/CURSOR-Y, Länge= SCREENWDITH)

Beispiel:

```
curline= ZocGetScreen("LEFTTOFCURSOR")
SAY "^M"
SAY "Der Text links vom Cursor war: "||curline
```

Beispiel:

```
width= ZocGetInfo("SCREENWIDTH")
posy= ZocGetInfo("CURSOR-Y")

data= ZocGetScreen(0,0, posy*width)
IF POS("A", data)=0 THEN DO
    SAY "Der Bildschirmabschnitt über dem Cursor enthält kein 'A'."
END
```

Beispiel:

```

cx= ZocGetInfo("SCREENWIDTH")
cy= ZocGetInfo("SCREENHEIGHT")

-- Schleife über alle Zeilen am Bildschirm
DO y= 0 TO cy-1
    line= ZocGetScreen(0,y, cx)
    -- Prüfe jede Zeile auf den Text "***SUCCESS**"
    IF POS("***SUCCESS**", line)>=1 THEN DO
        Call ZocMessageBox "***SUCCESS* gefunden in Zeile "||y
    END
END
END

```

Siehe auch: `ZocGetInfo("CURSOR-X")`, `ZocGetInfo("CURSOR-Y")`,
`ZocGetInfo("SCREENWIDTH")`, `ZocGetInfo("SCREENHEIGHT")`

ZocGetSessionOption(<schlüsselwort>)

Liefert die Zeichenkette Schlüsselwort-Wert für eine ZOC-Einstellung aus dem Verbindungsprofil (weitere Details hierzu beim Befehl [ZocSetSessionOption](#)).

Beispiel:

```

pair= ZocGetSessionOption("Beep")
PARSE VALUE pair WITH key="value"
IF value=="no" THEN DO
    SAY "Die Option Beep ist ausgeschaltet"
END

```

Beispiel:

```

pair= ZocGetSessionOption("EnqString")
PARSE VALUE pair WITH key="'value'"
SAY "Die eingestellte Enquire-Antwort lautet: "||value

```

Siehe auch: [ZocGetProgramOption](#), [ZocSetSessionOption](#), [ZocSetProgramOption](#)

ZocGlobalStore(<unterkommando>[, <options>])

Diese Gruppe von Befehlen erlaubt es, Werte über das Schließen und Neustarten von ZOC hinweg permanent zu speichern. Hierzu dient intern eine Datei als Datenpool. Die möglichen Operationen basieren auf den folgenden Unterbefehlen:

SELECT <name>

Wählt einen neuen Datenpool. Alle Namen, außer dem speziellen Namen `VOLATILE`, werden zur Bildung eines Dateinamens in der Form `<name>Global.ini` verwendet und er kann auch eine Pfadangabe enthalten, z.B. `C:\data\MyTestPool` (daraus wird dann auch die Datei `MyTestPoolGlobal.ini`).

Der Name `VOLATILE` dagegen wählt einen speziellen Datenpool an, der nur im Speicher gehalten wird und auf den alle ZOC Skripte Zugriff haben. Er verliert jedoch seine Werte, sobald das letzte ZOC-Fenster geschlossen wird.

INIT

Alle Werte im Datenpool löschen

GET <name>

Den Wert einer bestimmten Variable im Pool auslesen.

SET <name>, <value>

Einen Wert unter bestimmtem Namen speichern. Gibt `##OK##` oder einen Fehler zurück (s.u.).

PUT <name>, <value>

Identisch mit SET.

Konkurrierender Zugriff auf den Datenpool wird mit einer Lock-Datei abgesichert. Dies funktioniert auch über mehrere Computer hinweg, wenn der Datenpool auf einem Netzwerklaufwerk liegt. Rückgabewerte für die Befehle `GET`, `SET` und `PUT` sind `##OK##`, `##LOCKERROR##` und `##FILEERROR##`. Diese geben z.B. Aufschluss darüber, ob z.B. eine Schreiboperation erfolgreich war.

Beispiel:

```
CALL ZocGlobalStore "SELECT", "PermStore"
x= ZocGlobalStore("GET", "LASTX")
...
CALL ZocGlobalStore "SET", "LASTX", x

ret= ZocGlobalStore("SET", "VAL", "OK")
IF ret\="##OK##" THEN SAY "Could not set value"
```

ZocKeyboard(<befehl>[, <timeout>])

Diese Funktion erlaubt es einem REXX-Skript, Tastatureingaben in das Terminalfenster abzufragen. Es existieren die Unterbefehle `LOCK`, `UNLOCK` und `GETNEXTKEY`.

LOCK

Sperrt die Tastatur und verhindert Eingaben durch den Benutzer.

UNLOCK

Entsperrt die Tastatur.

GETNEXTKEY

Wartet auf den nächsten Tastendruck. Hierbei kann zusätzlich ein Timeout in Sekunden angegeben werden).

GETNEXTKEY liefert eine Zeichenkette in der Form `char|scancode|shift|ctrl|alt` zurück.

`char`: Eine Zwei-Byte Hexnummer, die den ASCII-Code des eingegebenen Zeichens repräsentiert.

`scancode`: Der physikalische Scancode, der von der Tastatur zurückgeliefert wird (anhand diesem Code können Tasten wie Pos1, Entf, F1, F2 usw. abgefragt werden).

`shift`, `ctrl` und `alt` liefern entweder 0 oder 1 zurück, je nachdem, ob die jeweilige Taste

zusammen mit der abgefragten Taste gedrückt wurde.

Das nachfolgende Beispiel zeigt, wie das Ergebnis von `ZocKeyboard()` in seine einzelnen Werte aufgeteilt werden kann und gibt die Ergebnisse auf dem Bildschirm aus.

Beispiel:

```
CALL ZocKeyboard "UNLOCK"
ret= ZocKeyboard("GETNEXTKEY")
PARSE VALUE ret WITH hexkey|"scan"|"shift"|"ctrl"|"alt"
key= X2C(hexkey)
SAY "Sie haben folgende Taste gedrückt hex/key: "hexkey"/"key"
SAY "Scan Code: "scan
SAY "Shift/Ctrl/Alt Status: "shift"/"ctrl"/"alt"
```

ZocLastLine()

Diese Funktion liefert die aktuelle Textzeile, die beim letzten `ZocWait/ZocWaitMux/ZocWaitLine` Befehl aufgelaufen ist (eine Alternative zu `ZocLastLine` ist die Funktion [ZocReceiveBuf](#)).

Beispiel:

```
CALL ZocSend "ATZ^M"
timeout= ZocWaitMux("OK", "ERROR")
IF (timeout\=640) THEN DO
    IF ZocLastLine()\="OK" THEN SIGNAL error
    CALL ZocConnect "555 3456"
END
```

Hinweis: In vielen Fällen wird stattdessen die Verwendung [ZocReceiveBuf](#) die flexiblere Lösung darstellen.

Siehe auch: Das Beispiel bei [ZocWaitLine](#)

ZocListFiles(<verz\maske> [, <trennzeichen>])

Die `ZocListFiles` Funktion liefert die Namen der Dateien, die in einem Verzeichnis gespeichert sind, entsprechend einer Maske aus DOS Jokerzeichen (z.B. `"c:\data\*.*)"`). Die Funktion liefert eine Zeichenkette, die die Anzahl der Dateien und die einzelnen Dateinamen, jeweils durch ein Leerzeichen getrennt, enthält, z.B. `"3 download.zip sales.txt foobar.fil"`. Dann kann mit Hilfe der REXX WORD Funktion einfach auf die einzelnen Namen zugegriffen werden (siehe Beispiel).

Falls Dateinamen selbst Leerzeichen enthalten, kann ein anderes Trennzeichen für die Liste der Dateinamen benutzt werden, z.B. würde die Funktion mit Trennzeichen `"|"` das Ergebnis `"3 download.zip|sales.txt|foobar.fil"` liefern. In diesem Fall kann die Zerlegung dann durch die Funktion `ZocString("PART", purelist, i, "|")` erfolgen.

Hinweis: Die Anzahl der Dateinamen ist auf 128 begrenzt, die max. Länge der Zeichenkette auf

4096 Bytes.

Beispiel:

```
files= ZocListFiles("C:\TEMP\*")

howmany= WORD(files, 1)
SAY "Anz. Dateien:" howmany

purelist= SUBSTR(files, LENGTH(howmany)+2)
DO i=1 TO howmany
    SAY "Datei " i "=" WORD(purelist, i)
END
```

Siehe auch: [ZocFilename](#), [ZocGetFilename](#), [ZocGetFileNames](#), [ZocGetFolderName](#), [ZocString](#)

ZocLoadKeyboardProfile [<keyfile>]

Lädt und aktiviert ein Tastaturprofil aus der entsprechenden Datei (*.zky).

Beispiel:

```
CALL ZocLoadKeyboardProfile "Alternate.zky"
```

ZocLoadSessionProfile <optsdatei>

Lädt und aktiviert ein Verbindungsprofilprofil aus der entsprechenden Datei (*.zoc).

Beispiel:

```
CALL ZocLoadSessionProfile "vt102opt.zoc"
```

ZocLoadTranslationProfile [<keyfile>]

Lädt und aktiviert ein Zeichenübersetzungstabelle aus der entsprechenden Datei (*.ztr).

Beispiel:

```
CALL ZocLoadTranslationProfile "7bitGerman.ztr"
```

ZocMath(<funktion>, <argument>[, <argument2>])

ZocMath berechnet die mathematische Funktion des zweiten Arguments. Gültige Funktionen sind [sin](#), [cos](#), [tan](#), [asin](#), [acos](#), [sqrt](#), [todeg](#), [torad](#), [bitand](#), [bitor](#), [bitxor](#).

Beispiel:

```
angle = 270
anglerad = ZocMath("torad", angle)
sinresult = ZocMath("sin", anglerad)
lowbits = ZocMath("bitand", 175, 15)
hibits = ZocMath("bitand", X2D(A5), X2D(F0))
```

ZocMenuEvent <menütext> [, <datei>]

Führt einen Befehl aus dem ZOC-Menü aus. Der <menütext> ist der Text wie im ZOC-Menü angezeigt, <datei> ist ein optionaler Dateiname, den manche Menübefehle übernehmen, statt den Benutzer nach einer Datei zu fragen.

Beispiel:

```
CALL ZocMenuEvent "Einfügen (kein Zeilenwechsel)"
CALL ZocMenuEvent "REXX Skript bearbeiten", "test.zrx"
```

ZocMessageBox(<text> [, <modus>])

Öffnet ein Nachrichtenfenster mit dem angegebenen Text (^M im Text erzeugt einen Zeilenwechsel).

Normalerweise wird ein Info-Fenster mit einem OK-Knopf (Modus 0) angezeigt. Modus 1 zeigt ein Fehler-Fenster mit einem OK-Knopf. Modus 2 zeigt eine Nachricht mit einem JA und NEIN-Knopf.

Der Rückgabewert ist entweder ##OK##, ##YES## oder ##NO##.

Beispiel:

```
CALL ZocMessageBox "Verbindung fehlgeschlagen!", 1
ret= ZocMessageBox("Fehler beim Speichern.^M^MNochmal versuchen?", 2)
IF ret=="##YES##" THEN DO
    ...
END
```

Siehe auch: [ZocAsk](#), [ZocDialog](#), [ZocAskFilename](#), [ZocAskFoldername](#), [ZocRequest](#), [ZocRequestList](#)

ZocNotify <text> [, <dauer>]

Zeigt eine Nachricht in einem kleinen Infofenster in der Mitte des Bildschirms an. Der zweite Parameter bestimmt ggf. die Zeit die das Fenster sichtbar ist (in Millisekunden).

Beispiel:

```
CALL ZocNotify "Hello World!", 1500
```

ZocPing(<ip-or-hostname>, <timeout>)

Sendet eine ICMP-Anfrage (ping) an den angegebenen Host und kehrt zurück, wenn er entweder eine Antwort erhält, oder wenn das Timeout (in Millisekunden) abgelaufen ist. Der Befehl gibt entweder die Zeichenfolge ##ERROR## oder die Roundtrip-Zeit in Millisekunden zurück.

Beispiel:

```
pingrc= ZocPing("www.emtec.com", 2500)
IF pingrc\="##ERROR##" THEN SAY "Received reply within "||pingrc||" ms"
```

ZocPlaySound <datei>

Spielt eine .WAV Datei ab.

Beispiel:

```
CALL ZocPlaySound "dingdong.wav"
```

ZocReceiveBuf(<puffergröße>)

Diese Funktion sammelt Teile der während einer Verbindung empfangenen Daten in einem Puffer und liefert den vorherigen Inhalt (falls vorhanden) als Zeichenkette zurück.

Normalerweise hat der Puffer die Größe Null, d.h. es werden keine Daten gesammelt. Um eine Datensammlung zu starten, muss die Funktion mit einer beliebigen Puffergröße aufgerufen werden. Danach werden die empfangenen Daten im Puffer gespeichert, bis entweder der Puffer voll ist oder bis die Funktion das nächste Mal aufgerufen wird. Der Folgeaufruf liefert den Pufferinhalt, löscht ihn und setzt ihn auf eine neue Größe.

Eine Aufrufsequenz der Funktion ZocReveiceBuf() am Beispiel einer Datenbankabfrage könnte etwa so aussehen:

Beispiel:

```
/* Puffergröße auf 256 Byte setzen */
CALL ZocTimeout 60
CALL ZocReceiveBuf 256
CALL ZocSend "lesen inhalt^M"
CALL ZocWait "Eingabe>"

/* die empfangenen Daten in Variable 'inhalt' schieben */
/* und den Puffer für Detail neu initialisieren */
inhalt= ZOCRECEIVEBUF(4096)
CALL ZocSend "lesen detail^M"
CALL ZocWait "Eingabe>"

/* die empfangenen Daten in Variable 'detail' speichern */
/* und den Puffer auf Null setzen */
detail= ZocReceiveBuf(0)

/* Hinweis: An diesem Punkt beginnen beide Zeichenketten
(inhalt und detail) immer mit dem Wort "lesen" und enden mit
den Zeichen "Eingabe>". */
```

Beispiel:

```
/* Remote-Umgebungsvariablen lesen und den TERM= Wert extrahieren */
Call ZocReceiveBuf 2048
Call ZocSend "set^M"

/* auf das aktuelle (!) Prompt warten */
Call ZocWait "PROMPT: ~username$"
data= ZocReceiveBuf(0)
```

```
/* Google kennt weitere Details zu "REXX PARSE COMMAND" und  
   den PARSE-Befehl, der hier zum Extrahieren der Daten verwendet wird */  
PARSE VALUE data WITH ."TERM="term .
```

```
SAY "Der TERM Wert der Gegenstelle ist " term
```

Hinweis: Per DDE muss der Befehl unbedingt per DdeRequest und nicht per DdeExecute ausgeführt werden.

Siehe auch: [ZocLastLine](#), [ZocWait](#), [ZocWaitLine](#), [ZocWaitForSeq](#), [ZocString\("LINE" ...\)](#)

ZocRegistry(<befehl>[, <options>])

Diese Gruppe von Funktionen erlaubt den Zugriff auf die Windows-Registry.

OPEN <basekey>, <name>

Gibt einen <hkey> Handle zum Zugriff auf einen Teil der Registry zurück oder im Fehlerfall **###ERROR##**. Der <basekey> kann entweder [HKEY_CURRENT_USER](#) oder [HKEY_LOCAL_MACHINE](#) sein.

OPEN liefert einen so genannten hkey als Returnwert, der dann für Schreiben und Lesen in diesem Teil der Registry verwendet werden kann.

WRITE <hkey>, <value>, <data>

Schreibt <data> in den mit <hkey> angegebenen Teil der Registry. Falls <data> im Format ["DWORD:nnnn"](#) angegeben wird, wird der darin angegeben Dezimalwert nnnn dezimal als REG_DWORD eingetragen. Falls <data> in die Form ["BINARY:xxxxxx..."](#) hat, wird die darin angegeben Hex-Zeichenkette in Bytes umgewandelt und als REG_BINARY eingetragen. Ansonsten wird <data> als Typ REG_SZ (Zeichenkette) gespeichert.

READ <hkey>, <value>

Liest einen Wert aus dem <hkey>-Teil der Registry. Falls der Wert dort als REG_DWORD gespeichert ist, wird ["DWORD:nnnn"](#) zurückgeliefert. Beim Typ REG_BINARY wird das Ergebnis in der Form ["BINARY:xxxx..."](#) angegeben (xxxx ist dabei eine Folge von Hex-Ziffern, die z.B. mit [x2c](#) in Bytes zurückverwandelt werden kann). Für Typ REG_SZ liefert die Funktion die in der Registry gespeicherte Zeichenkette.

ENUM <hkey>, <n>

Liefert den <n>. Namen aus <hkey>.

TEST <hkey>, <value>

Diese Funktion prüft, ob der angegebene Wert existiert und liefert entweder **###ERROR##**, oder eine Zeichenkette in der Form **##OK## TYPE nt LENGTH nl**, bei der nt und nl dezimale Werte sind, die den Typ und die Länge des Registry-Eintrags angeben.

CLOSE <hkey>

Beendet den Zugriff auf <hkey>.

Beispiel:

```
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 x
END

CALL ZocRegistry "CLOSE", hk
EXIT

```

ZocRequest(<titel>, <opt1> [, <opt2> [, <opt3>]])

Öffnet einen Dialog mit mehreren Knöpfen und liefert eine Zeichenkette mit dem Namen des ausgewählten Knopfes zurück.

Beispiel:

```

answer= ZocRequest("Was möchten Sie?", "Milch", "Honig")
IF answer=="Milch" THEN DO
  ...
END

```

Siehe auch: [ZocAsk](#), [ZocDialog](#), [ZocAskFilename](#), [ZocRequestList](#)

ZocRequestList(<title>, <opt1> [, ...])

Öffnet ein Fenster mit einer Liste von Werten und gibt den Index der Auswahl (oder -1 für Abbruch) zurück. Falls nur eine Option übergeben wird, wird diese als Liste von Optionen, die durch senkrechte Striche getrennt sind, interpretiert.

Beispiel:

```

answer= ZocRequestList("Bitte wählen!", "Bier", "Wein", "Whiskey",
  "Schnaps")
IF answer=3 THEN DO
  ...
END

answer= ZocRequestList("Bitte wählen!", "Bier|Wein|Whiskey|Schnaps")
IF answer=3 THEN DO
  ...
END

```

Siehe auch: [ZocAsk](#), [ZocDialog](#), [ZocAskFilename](#), [ZocRequest](#), [ZocRequestList](#)

ZocRespond <text1> [, <text2>]

Sendet automatisch *text2*, wenn während der Bearbeitung von ZocDelay oder ZocWait Befehlen der Text *text1* empfangen wird.

Maximal 64 ZocRespond Befehle können zur gleichen Zeit aktiv sein. <text1> darf weder CR noch LF enthalten.

Wird nur *text1* angegeben und kein zweiter Parameter, oder ist der zweite Parameter leer, so wird ein bestehender ZocRespond für *text1* gelöscht. Wenn *text1* leer ist (""), werden alle ZocRespond Einträge gelöscht.

Beispiel:

```
/* Auf 'Befehl' warten und alle Abfragen dazwischen mit Enter bestätigen */
CALL ZocRespond "Enter", "^M"
CALL ZocRespond "Mehr", "^M"
timeout= ZocWait("Eingabe>")
/* Responder löschen */
CALL ZocRespond "Enter"
CALL ZocRespond "Mehr"
```

Das obige Beispiel wartet, bis der Text *Eingabe>* empfangen wird. Während der Wartezeit werden alle *Enter* und *Mehr* Abfragen automatisch mit Enter beantwortet. Nach dem Empfang von "Befehl" werden die Respond-Befehle wieder gelöscht.

ZocSaveSessionProfile [<dateiname>]

Speichert die aktuellen Einstellungen des Verbindungsprofils (siehe [Optionen→Verbindungsprofil](#)) in einer Datei. Wenn kein Dateiname angegeben wird, öffnet ZOC ein Dateiauswahlfenster.

Beispiel:

```
CALL ZocSetSessionOption "JumpScroll=3"
CALL ZocSaveSessionProfile "Fastscroll.zoc"
```

ZocSend <text>

Sendet den angegebenen Text zur Gegenstelle. Die Zeichen werden hierzu programmintern wie eine Folge von Tastendrücken verarbeitet. Die Verzögerung hierbei wird durch die Text-Sende Optionen in [Optionen→Verbindungsprofil→Text Senden](#) gesteuert. Falls Sie schnelleren direkteren Versand der Daten benötigen, können Sie ZocSendRaw benutzen.

Außerdem werden in nahezu allen Emulationen wie ^M automatisch in die üblichen echten Werte umgewandelt (siehe [Steuerzeichen](#)). Ausnahmen sind die TN3270/5250 Emulationen, dort gilt ^M als *Newline/FieldExit*, ^I als *Tab* und ^Z als *Datenfreigabe/Enter*.

Beispiel:

```
CALL ZocSend "JOE USER^M" /* send JOE USER<enter>*/
```

Beispiel:

```
/* Unix login Sequence */  
CALL ZocWait "login:"  
CALL ZocSend "harry^M"  
CALL ZocWait "password:"  
CALL ZocSend "alohomora^M"
```

Beispiel:

```
/* 3270/5250 example */  
Call ZocSetCursorPos 12,4  
CALL ZocSend "Freddie"  
CALL ZocSendEmulationKey "NewLine"  
CALL ZocSend "Elm Street"  
CALL ZocSendEmulationKey "Enter"  
  
/* 3270/5250 same as above, using  
   emulation specific control codes */  
CALL ZocSend "Freddie^MElm Street^Z"
```

ZocSendEmulationKey <tastenbezeichnung>

Simuliert das Drücken einer Sondertaste der Emulation, z.B. senden der Taste [F17](#) der VT220 Emulation oder senden der [Attn](#) unter TN3270.

Erlaubte Werte und deren tatsächliche Funktion hängen von der momentan aktiven Terminalemulation ab.

Die Tastenbezeichnungen selbst sind im Anhang unter [Tastenbezeichnungen](#) nach Emulation gruppiert aufgelistet.

Beispiel:

```
CALL ZocSetEmulation "VT100"  
CALL ZocSendEmulationKey "PF1"
```

ZocSendRaw <daten>

Dieses Kommando verschickt die angegebenen Daten ohne sie irgendwie zu interpretieren (d.h. ohne sie z.B. auf Steuercodes wie `^M` zu untersuchen). Für die Umwandlung von Steuercodes müssen stattdessen die REXX Zeichenkettenfunktionen wie z.B. `X2C(0D)` oder `ZocCtrlString("^M")` verwendet werden.

Mit `ZocSendRaw` können daher jegliche binäre Daten verschickt werden, z.B. `hex 42 01 00 05 41 43`.

Beispiel:

```
CALL ZocSendRaw "Login"||X2C(0d) /* Login<enter> */  
  
/* Dreimal das gleiche Ergebnis: */  
CALL ZocSendRaw X2C(420100054143)  
CALL ZocSendRaw "B"||X2C(01)||X2C(00)||X2C(05)||"AC"  
CALL ZocSendRaw ZocCtrlString("B^A^@^EAC")
```

Siehe auch: [ZocSend](#), [ZocSendEmulationKey](#), [ZocCtrlString](#)

ZocSessionTab(<unterkommando>, <parameter>)

Diese Funktion erlaubt es, die Registerblätter mit Verbindungen zu manipulieren. Der <unterkommando> Parameter legt die durchzuführende Operation fest, die weiteren Parameter hängen vom Unterkommando ab:

CLOSEATEXIT

Schließt das aktuelle Registerblatt sobald das Skript beendet ist.

Beispiel: `CALL ZocSessionTab "CLOSEATEXIT"`

CLOSETAB

Schließt das Registerblatt mit dem angegebenen Index (0 für das Blatt ganz links, -1 für das aktive).

Beispiel: `CALL ZocSessionTab "CLOSETAB", 2`

GETCOUNT

Liefert die Anzahl der Registerblätter mit Verbindungen.

Beispiel: `howmany= ZocSessionTab("GETCOUNT")`

GETCURRENTINDEX

Liefert den index des Registerblatts, in dem das Skript läuft.

Beispiel: `myidx= ZocSessionTab("GETCURRENTINDEX")`

GETINDEXBYNAME, <name>

Liefert den Index des ersten Registerblatts, das den angegebenen Namen hat, oder -1 für keines.

Beispiel: `srvidx= ZocSessionTab("GETINDEXBYNAME", "My Server")`

GETNAME, <index>

Liefert den Name des Registerblattes mit dem gegebenen Index (0 für das Blatt ganz links). Ein Index von -1 bezieht sich auf das Registerblatt in dem das Skript läuft.

Beispiel: `name= ZocSessionTab("GETNAME", -1)`

ISCONNECTED, <index>

Liefert als Ergebnis `##YES##` oder `##NO##` je nachdem ob das Registerblatt mit dem angegebenen Index momentan mit einer Gegenstelle verbunden ist.

Beispiel: `yesno= ZocSessionTab("ISCONNECTED", 2)`

NEWSESSION, <titel>, <aktiv>, <verbindungsprofil>[, <verbindung>, <skript>]

Erstellt eine neue Session und liefert den neuen Session-Index zurück.

Parameters:

<titel>: Eine Zeichenkette mit dem Namen für das neue Registerblatt.

<aktiv>: 1 oder 0, je nachdem ob das neue Blatt in den Vordergrund geholt werden soll oder nicht.

<verbindungsprofil>: Name eines Session-Profiles (z.B. `MyProfile.zoc`) oder `##NULL##` für Standard.

<verbindung>: Eine Zeichenkette für den Verbindungsaufbau wie beim [CONNECT-Kommandozeilenparameter](#) beschrieben oder der Text `CALL`: gefolgt vom Namen eines Eintrags aus dem Adressbuch oder `##NULL##` (keine Verbindung).

<skript>: Der Name eines REXX-Skripts das im neuen Registerblatt ausgeführt werden soll.

Beispiel:

```
idx1= ZocSessionTab("NEWSESSION", "Test", 1, "##NULL##", "CALL:Mein  
Server")
```

```
idx2= ZocSessionTab("NEWSESSION", "Test2", 0, "SSHProfil.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>, <menü>

Sendet Text an die Verbindung mit dem angegebenen Index (siehe die Beschreibung für <index> bei GETNAME). Der Parameter <menü> entspricht dem Befehl [ZocMenuEvent](#).

Beispiel: `CALL ZocSessionTab "MENUEVENT", 2, "Verbindung beenden"`

RUNSCRIPT, <index>, <title>

Startet ein Skript im mit <index> angegeben Registerblatt (siehe die Beschreibung für <index> bei GETNAME). Beachten Sie, dass in dem Registerblatt nicht bereits kein Skript laufen sollte (das heißt auch, dass RUNSCRIPT für das aktuelle Blatt in dem der Befehl selbst ausgeführt wird, keinen Sinn macht).

Beispiel: `CALL ZocSessionTab "RUNSCRIPT", 2, "configure.zrx"`

SEND, <index>, <text>

Sendet Text an die Verbindung mit dem angegebenen Index (siehe die Beschreibung für <index>

bei GETNAME).

Beispiel: `CALL ZocSessionTab "SEND", 2, "exit^M"`

SETBLINKING, <index>, <blinkflag>

Aktiviert oder deaktiviert das Blinken für das Register mit dem angegebenen Index (siehe die Beschreibung für <index> bei GETNAME). Das <blinkflag> ist entweder 1 oder 0.

Beispiel: `CALL ZocSessionTab "BLINKING", -1, 1`

SETCOLOR, <index>, <color>

Setzt die Farbe für das Register mit dem angegebenen Index (siehe die Beschreibung für <index> bei GETNAME). Die Farbe ist eine Nummer von 0 - 7.

Beispiel: `CALL ZocSessionTab "SETCOLOR", -1, 4`

SETNAME, <index>, <title>

Setzt den Titel für das Register mit dem angegebenen Index (siehe die Beschreibung für <index> bei GETNAME).

Beispiel: `CALL ZocSessionTab "SETNAME", -1, "This Session"`

SWITCHTO, <index>

Aktiviert das Register mit dem angegebenen Index (siehe die Beschreibung für <index> bei GETNAME).

Beispiel: `CALL ZocSessionTab "SWITCHTO", 2`

WRITE, <index>, <text>

Schreibt Text auf den Bildschirm der Session mit dem angegebenen Index (ähnlich als würde man den ZocWrite-Befehl in diesem Tab verwenden).

Beispiel: `CALL ZocSessionTab "WRITE", 1, "Fortschritt 10%^M"`

Beispiel:

```
/* 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 <dateiname>

Legt den Namen für die Audit-Logdatei fest. Das Audit-Log ist ein Protokoll der Kommunikation, das nicht vom Benutzer abgeschaltet werden kann; siehe auch den Eintrag in der ADMIN.INI Datei.

Mit "" (leere Zeichenkette) als Dateiname wird die Protokollierung deaktiviert.

ZocSetAutoAccept 1|0

Schaltet die automatische Annahme von eingehenden Verbindungen ein oder aus, vorausgesetzt die aktive Kommunikationsmethode unterstützt dies.

Beispiel:

```
CALL ZocSetDevice "Telnet"  
CALL ZocSetAutoAccept 1 /* accept calls */
```

ZocSetCursorPos <zeile>, <spalte>

Dieser Befehl bewegt den Cursor auf dem Bildschirm in einer TN3270 Emulation an die angegebene Zeile und Spalte (die Position links oben hat die Adresse 1/1).

Beispiel:

```
CALL ZocSetCursorPos 1,15  
Call ZocSendEmulationKey "Enter"
```

ZocSetDevice <name> [,<verbindungstyp-parameter>]

Wechselt die Verbindungsmethode (Verbindungstyp). Als Name kann eine der in [Verbindungsprofil→Verbindungstyp](#) verfügbaren Methoden angegeben werden.

Der optionale Verbindungstyp-Parameter ist ein Text, der die Einstellungen enthält, mit denen der Verbindungstyp geöffnet werden soll, z.B. bei Telnet die CR/NUL Einstellungen, usw. Die Funktion [ZocSetDeviceOpts](#) beschreibt, wie gültige Verbindungstyp-Parameter abgefragt werden können. Falls keine Parameter angegeben werden, werden die in der Methode eingestellten Optionen des momentan aktiven Verbindungsprofils verwendet.

Beispiel:

```
CALL ZocSetDevice "TELNET"  
CALL ZocConnect "bbs.channel11.com"
```

Beispiel:

```
CALL ZocSetDevice "SERIAL/MODEM", "[1]COM3:57600-8N1|9|350"
```

ZocSetDeviceOpts <parameter-string>

Dies ist eine sehr fortgeschrittene Funktion, die erlaubt, Einstellungen für die Verbindungsmethode (Verbindungstyp) direkt von REXX aus zu ändern. Allerdings ist der Parametertext für Verbindungsmethoden nicht standardisiert, d.h. um den entsprechenden Parametertext zu finden, müssen die entsprechenden Optionen zunächst manuell im Dialog eingestellt und der den Optionen entsprechende Parametertext dann abgefragt werden.

Angenommen, es soll eine Modem-Verbindung auf COM3, mit 57600 Baud, RTS/CTS, gültigem Carrier und einer Break-Zeit von 350ms gestartet werden.

1. Nach Optionen→Verbindungsprofil→Verbindungstyp gehen und unter Serial/Modem diese Option einstellen
2. Das Verbindungsprofil-Fenster schließen
3. Shift+Strg+F10 drücken
4. Der Parametertext wird als "Device-Options" ausgegeben, in diesem Fall [\[1\]COM3:57600-8N1|9|350](#).
5. Dieser Text kann als Parameter-String für ZocSetDeviceOpts verwendet werden.

Beispiel:

```
/* Serielle Parameters setzen:
   COM3, 57600-8N1, RTS/CTS, Valid-CD, 350ms-Break */
CALL ZocSetDeviceOpts "[1]COM3:57600-8N1|9|350"
```

Beispiel:

```
/* Telnet auswählen und Optionen auf
   "Mit lokalem echo starten" setzen */
CALL ZocSetDevice "TELNET"
CALL ZocSetDeviceOpts "[3]12"
```

ZocSetEmulation <emulationname>[, <emuparm-string>]

Mit dem Befehl ZocSetEmulation kann skriptgesteuert eine andere Emulation aktiviert werden. Als Parameter kommen alle Namen in Frage, die im Verbindungsprofil als Emulation angezeigt werden, z.B. VT220, TN3270, etc.

Beispiel:

```
CALL ZocSetEmulation "Xterm"
```

ZocSetHostEntry "name", "<schlüsselwort>=<wert>"

Setzt einen neuen Wert in einen Adressbucheintrag. Alle möglichen Kombinationen für die Schlüsselwörter können Sie sehen, wenn Sie die Datei [HostDirectory.zhd](#) im ZOC Datenordner in einen Editor laden.

Wenn im zweiten Parameter statt Schlüsselwort und Wert die Zeichenkette "##NEW##" angegeben

wird, erstellt der Befehl einen neuen leeren Adressbucheintrag mit diesem Namen (falls noch keiner existiert).

Beispiel:

```
CALL ZocSetHostEntry "ZOC-Support-Mailbox", "emulation=1"

pair= ZocSetHostEntry("ZOC-Support-Mailbox", "calls")
PARSE VALUE pair WITH key="value"
value= value+1
CALL ZocSetHostEntry "ZOC-Support-Mailbox", "calls="||value

/* Hinweis: Anführungszeichen unten beachten! */
value= "3781302"
CALL ZocSetHostEntry "ZOC Support BBS", 'connectto="3781302"'
CALL ZocSetHostEntry "ZOC Support BBS", 'connectto="'||value||'"'
```

Beispiel:

```
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"
```

Siehe auch: [ZocSetSessionOption](#), [ZocGetHostEntry](#)

ZocSetLogfileName <name>

Ändert den Namen der Logdatei für die aktuelle Session.

Beispiel:

```
CALL ZocSetLogfileName "Heute.log"
```

ZocSetLogging 0|1 [, 1]

Schaltet die Protokollierung der eingehenden Daten in einer Datei ein oder aus. Wenn ein zweiter Parameter mit Wert 1 angegeben wird, wird das kleine Statusfenster in der Mitte des Bildschirms unterdrückt.

Beispiel:

```
CALL ZocSetLogging 1
```

ZocSetMode <key>, <wert>

Dieser Befehl erlaubt es bestimmte Verarbeitungen im REXX Prozessor zu beeinflussen. Folgende key-Werte werden unterstützt:

SAY

Ein Wert von [RAW](#) steuert, dass der SAY Befehl SteuerCodes wie ^M nicht auflöst. [COOKED](#) schaltet zurück auf das Standardverhalten.

Beispiel:

```
CALL ZocSetMode "SAY", "RAW"
```

RESPOND

Ein Wert von [RAW](#) steuert, dass der ZocRespond Befehl SteuerCodes Codes wie ^M nicht auflöst (für beide ZocRespond-Parameter).

Beispiel:

```
CALL ZocSetMode "RESPOND", "RAW"
```

ZocSetProgramOption "<schlüsselwort>=<wert>"

Dieser Befehl ändert eine Einstellung aus dem Fenster [Optionen→Programmeinstellungen](#) von ZOC ([ZocSetSessionOption](#) verändert stattdessen eine Einstellung aus einem Verbindungsprofil). Die Funktion arbeitet wie [ZocSetSessionOption](#), allerdings basiert sie auf Wertepaaren aus der Datei [Standard.zfg](#). Die Schlüsselnamen dort sind auch im wesentlichen selbsterklärend oder sie können die Funktion zur Aufzeichnung eines REXX-Skripts aktivieren, die Änderung in den Programmeinstellungen vornehmen und sich dann das aufgezeichnete Skript ansehen..

Beispiel:

```
CALL ZocSetProgramOption "SafAskClrCapt=yes"
CALL ZocSetProgramOption 'SkriptPath="ZocREXX"' /* Anführungszeichen
beachten */
CALL ZocSetProgramOption 'SkriptPath="||pathvar||"' /*
Anführungszeichen beachten */
```

Siehe auch: [ZocCommand\("SAVEPROGRAMSETTINGS"\)](#), [ZocGetProgramOption](#), [ZocGetSessionOption](#), [ZocSetSessionOption](#)

ZocSetScriptOutputDestination "DEFAULT|DATASTREAMBROWSER|FILE:<filename>"

Dieser Befehl leitet die Ausgabe von REXX Befehlen vom Bildschirm (DEFAULT) entweder in das Datenstromfenster (Anzeigen-Menü) oder in eine Datei um. Dies betrifft Ausgaben der REXX Befehle [SAY](#) und [TRACE](#). Ausgaben von [ZocWrite](#) erscheinen weiterhin im Terminal-Bereich.

Beispiel:

```
CALL ZocSetScriptOutputDestination "DATASTREAMBROWSER"
```

Siehe auch: [ZocWrite](#), [ZocWriteln](#)

ZocSetSessionOption "<schlüsselwort>=<wert>"

Setzt eine beliebige ZOC-Option aus dem Fenster [Optionen→Verbindungsprofil](#) anhand eines

Schlüsselwort-Wert Paares. Die möglichen Wertepaare sind in der Datei [Standard.zoc](#) (und allen anderen Verbindungsprofildateien) enthalten und stellen die jeweilige Konfiguration dar.

Um herauszufinden, welche Option zu welchem Schlüsselwort gehört, genügt es, die entsprechenden Einstellungen in Verbindungsprofildialog von ZOC vorzunehmen, abzuspeichern und die veränderte Schlüsselwort-Wert Zeile zu suchen. Alternativ können Sie auch die Funktion zur Aufzeichnung eines REXX-Skripts aktivieren, die Änderung am Verbindungsprofil vornehmen und sich dann das aufgezeichnete Skript ansehen.

Beispiel:

```
CALL ZocSetSessionOption "Beep=no"
CALL ZocSetSessionOption "JumpScroll=3"
CALL ZocSetSessionOption "ShowChat=no"
CALL ZocSetSessionOption 'MdmIni="ATZ^M"' /* Anführungszeichen beachten
*/
CALL ZocSetSessionOption 'TransAutoRemove="||valvar||"' /*
Anführungszeichen beachten */
```

Hinweis: ZocSetSessionOption/ZocGetSessionOption funktioniert nur für Optionen aus Optionen→Verbindungsprofil. Einträge aus den Programmeinstellungen können stattdessen mit [ZocSetProgramOption](#) bearbeitet werden.

Siehe auch: ZocCommand("SAVESESSIONPROFILE"), [ZocSaveSessionProfile](#), [ZocGetProgramOption](#), [ZocSetProgramOption](#), [ZocGetSessionOption](#)

ZocSetTimer <hh:mm:ss>

Setzt den Timer auf den angegebenen Wert. Mit einer leeren Zeichenkette als Parameter, wird als Ergebnis die Zeit des Timers in Sekunden zurückgeliefert. Ein Aufruf der Funktion mit Parameter "STOP" hält den Timer an und "RESUME" zählt einen angehaltenen Timer weiter.

Beispiel:

```
CALL ZocSetTimer "00:00:20"
```

ZocSetUnattended 0|1

Schaltet den Modus für unbeaufsichtigten Betrieb ein oder aus (siehe [Kommandozeilen-Parameter /U](#)).

Beispiel:

```
CALL ZocSetUnattended 1
```

ZocShell <befehl>, [<anzeigemodus>]

Führt ein Programm per `cmd.exe /c <command>` (Windows) oder `/bin/bash -c "<command>"` (macOS). Dieser Befehl ist ähnlich dem REXX-Befehl `ADDRESS CMD "<befehl>"` und erlaubt im Prinzip die Ausführung eines jeden Befehls, der auch im Shell-Fenster des Betriebssystems eingegeben werden kann.

Der optionale Parameter für den Anzeigemodus steuert, wie das zugehörige schwarze Shell-Fenster (nur Windows) angezeigt werden soll:

0= normal, 1= versteckt, 2= minimiert, 3= maximiert.

Beispiel:

```
CALL ZocShell "DEL FILE.TMP"  
CALL ZocShell "touch /tmp/file.lock", 1
```

Siehe auch: [ZocShellExec](#), [ZocShellOpen](#), [ZocFileDelete](#), [ZocFileRename](#)

ZocShellExec <command> [, <viewmode>]

Übergibt ein Programm zur direkten Ausführung an das Betriebssystem (d.h. ohne den Interpreter des schwarze Shell-Fensters). Dies funktioniert unter Windows nur für .exe und .com Programme, nicht jedoch für .cmd-Dateien und interne Befehle wie DIR, REN, DEL, etc. (hierfür kann alternativ [ZocShell](#) verwendet werden).

Der optionale Anzeige-Parameter steuert, wie das Fenster des Programms angezeigt werden soll:
0= normal, 1= versteckt, 2= minimiert, 3= maximiert.

Beispiel:

```
CALL ZocShellExec 'notepad.exe "neuer.txt"'
```

Siehe auch: [ZocShell](#), [ZocShellOpen](#)

ZocShellOpen <dateiname>

Übergibt eine Datei zum Öffnen an das Betriebssystem. Diese Funktion entspricht in etwa dem Doppelklick auf einer Datei und öffnet die Datei mit dem Programm, das im Betriebssystem für diesen Dateityp registriert ist (z.B. einen PDF-Viewer für PDF-Dateien oder Notepad für TXT-Dateien).

Alternativ kann statt eines Dateinamens auch ein URL angegeben werden.

Beispiel:

```
CALL ZocShellOpen 'C:\DOWNLOADS\Report.pdf'
```

Beispiel:

```
CALL ZocShellOpen 'https://www.emtec.com/'
```

Siehe auch: [ZocShell](#), [ZocShellExec](#)

ZocString(<unterkommando>, <inputstring>, <p1> [, <p2>])

Diese Funktion bearbeitet oder analysiert Zeichenketten entsprechend des Unterkommandos sowie der weiteren Parameter und liefert ein darauf basierendes Ergebnis zurück.

Unterkommandos:

LINE

Liefert das <p1>te Element von <inputstring> das durch ein Linefeed (hex 0A) getrennt ist (umgangssprachlich gesagt ist das die <p1>te Zeile). Vom Ergebnis werden am Anfang und Ende ggf. vorhandene Carriage-Return (hex 0D) entfernt. Diese Funktion ist hilfreich um mehrzeilige Ergebnisse von [ZocReceiveBuf](#) zu zerlegen, z.B. liefert `name=ZocString("LINE", recdata, 4)` die vierte Zeile aus der Variable recdata.

LINECOUNT

Anzahl der Element für LINE.

LOAD

Liefert den Inhalt der Datei mit Namen <inputstring> zurück (Die Datei wird im Textmodus geladen, Zeilenenden werden zu LF (hex 0A) konvertiert). Das LINE Unterkommando von ZocString kann verwendet werden, um Zeilen aus der zurückgelieferten Zeichenkette zu extrahieren.

SAVE

Speichert den Inhalt der Zeichenkette <p1> in eine Datei mit Namen <inputstring> (Die Datei wird als Textdatei gespeichert, LF Zeilenenden werden unter Windows zu CR/LF konvertiert).

MIME-ENCODE

Konvertiert <inputstring> nach base-64/MIME.

MIME-DECODE

Konvertiert <inputstring> zurück aus base-64/MIME.

UTF8-ENCODE

Konvertiert <inputstring> von 8-Bit Zeichen nach UTF8.

UTF8-DECODE

Konvertiert <inputstring> von einem UTF8-String in eine 8-bit Zeichenkette.

AES256-ENCRYPT

Verschlüsselt den String <p1> mit dem Kennwort aus <inputstring>. Das Ergebnis ist eine Zeichenkette im MIME-Format.

AES256-DECRYPT

Entschlüsselt den String aus <p1> (MIME-Format) mit dem Kennwort aus <inputstring>.

PART

Holt den <p1>ten Teil von <inputstring> der durch das Zeichen <p2> getrennt ist, z.B. liefert `name=ZocString("PART", "Anne|Charly|Joe", 2, "|")` den String "Charly".

PARTCOUNT

Liefert die Anzahl der durch <p1> getrennten Teile von <inputstring>, z.B. wird `count=ZocString("PARTCOUNT", "Anne|Charly|Joe", "|")` den Wert 3 liefern.

REPLACE

Liefert eine Kopie von <inputstring> in der alle <p1> durch <p2> ersetzt sind, z.B. `besser=ZocString("REPLACE", str, "HyperTerminal", "ZOC")`

REMOVE

Liefert eine Kopie von <inputstring> in der der String <p1> entfernt wurde, z.B. `besser=ZocString("REMOVE", str, "HyperTerminal")`

REMOVECHARS

Liefert eine Kopie von <inputstring> in der alle Zeichen von <p1> entfernt wurden, z.B. löscht `str=ZocString("REMOVECHARS", str, "0123456789"||X2C(09))` alle Ziffern und Tab-Zeichen aus der Variable str.

TAB

Gibt das <p1>te Element von <inputstring> zurück, das durch ein Tab-Zeichen getrennt ist, z.B. `spalte2= ZocString("TAB", tabbed_data, 2)`.

TABCOUNT

Anzahl der durch Tab-zeichen getrennten Elemente im <inputstring>.

WORD

Gibt das <p1>te Element von <inputstring> zurück, das durch ein Leerzeichen getrennt ist, z.B. bringt `name= ZocString("WORD", "The quick brown fox", 3)` als Ergebnis "brown".

WORDCOUNT

Anzahl der WORD Elemente im <inputstring>.

Beispiel:

```
CALL ZocReceiveBuf 1024
CALL ZocSend "ps -a^M"
CALL ZocWait "$" /* prompt after command output */
data= ZocReceiveBuf(0)

/* Anzeige der Befehlsausgabe Zeile fuer Zeile
   (jedoch ohne das Echo in der die ersten) */
howmany= ZocString("LINECOUNT", data)
DO i=2 TO howmany
  SAY ZocString("LINE", data, i)
END
```

Beispiel:

```
key= "Secret.740.$%&"
n= ZocString("AES256-ENCRYPT", key, "Hello World!")
SAY "Encoded: "||n
n2= ZocString("AES256-DECRYPT", key, n)
SAY "Decoded: "||n2
```

Siehe auch: [ZocCtrlString](#)

ZocSuppressOutput 0|1

Schaltet die Unterdrückung der Bildschirmausgabe ein oder aus. Dieser Befehl erlaubt das Senden/Empfangen von Daten ohne Bildschirmausgabe. Das Mitprotokollieren in den Puffer sowie in die Logdatei wird ebenfalls unterdrückt.

Das Unterdrücken der Bildschirmausgabe wird automatisch zurückgesetzt, wenn das Skript oder die Verbindung beendet ist.

ZocSyncTime <ms>

Da REXX in einem eigenen Thread ausgeführt wird, kann es vorkommen, dass eingehende Daten

schneller ankommen als sie vom REXX-Programm verarbeitet werden können, d.h. es ist möglich, dass Text, auf den ein REXX-Programm wartet, von ZOC bereits empfangen wurde, bevor REXX den ZocWait Befehl erreicht.

Ein typisches Beispiel für diese Problematik ist eine Schleife, um alle ankommenden Textzeilen anzuzeigen.

Beispiel:

```
DO FOREVER
  timeout= ZocWaitLine()
  IF timeout\=640 THEN DO
    zeile= ZocLastLine()
    /* Zeile wird hier verarbeitet */
  END
END
```

Im obigen Beispiel empfängt ZOC möglicherweise bereits wieder Text, während das REXX-Programm die Zeile noch verarbeitet und noch bevor es bereit ist für den nächste [ZocWaitLine](#) Befehl.

Um das zu verhindern, wartet ZOC nach gefundenem Text bis eine definierte Zeit (die Sync-Zeit) verstrichen ist, oder bis REXX ein weiteres Kommando ausführt das mit dem Hauptfenster interagiert (z.B. ZocSend, ZocWait, etc. sowie auch SAY, TRACE usw. weil auch diese in dem Hauptfenster Ausgaben machen).

Sollten diese 250ms zu kurz für das jeweilige REXX-Programm sein, kann diese Wartezeit mit ZocSynctime erhöht werden.

Wichtig: Da aber auch bei höherer Synchronisationszeit ein Befehl (etwa ZocWrite oder SAY) den Empfang weiterlaufen lassen kann, ist es empfehlenswert, die Daten zunächst nur zu sammeln und unnötige weitere Befehle zu vermeiden und erst danach in einem zweiten Schritt die Daten zu verarbeiten (Beim Befehl [ZocWaitLine](#) finden Sie ein Beispiel wie das aussehen kann). Alternativ können Sie statt mehrerer Wait-Befehle auch den Befehl [ZocReceiveBuf](#) verwenden um die Daten auf einmal abzugreifen (siehe das Beispiel dort).

Siehe auch: [ZocWait](#), [ZocWaitIdle](#), [ZocWaitLine](#), [ZocWaitMux](#), [ZocReceiveBuf](#), [ZocLastLine](#)

ZocTerminate [<return-code>]

Schließt das Fenster und beendet ZOC nachdem das REXX-Programm vollständig ausgeführt und beendet wurde. Normalerweise folgt dem Befehl ZocTerminate deshalb der REXX-Befehl `EXIT`.

Wenn der Return-Code Parameter angegeben wird, gibt ZOC diesen Wert an das Betriebssystem oder an das aufrufende Programm zurück.

Beispiel:

```
CALL ZocTerminate
EXIT
```

ZocTimeout <sek>

Setzt die Timeout-Zeit für ZocWait/ZocWaitMux/ZocWaitLine/ZocEventSemaphore.

Beispiel:

```
/* ZocWait wartet max. 30 Sek. auf Ergebnis und gibt danach auf */
CALL ZocTimeout 30

/* Warte bis Gegenstelle 'ready' sendet, oder bis die max. Timeout-Zeit
erreicht ist */
ret= ZocWait("ready")
IF ret\=640 THEN SAY "Gegenstelle bereit!"
ELSE SAY "System war nicht innerhalb 30 Sek. bereit"
```

Siehe auch: [ZocWait](#), [ZocWaitIdle](#), [ZocWaitLine](#), [ZocWaitMux](#), [ZocEventSemaphore](#), [ZocSyncTime](#)

ZocUpload <protokoll>[:<optionen>], <dateiname>

Startet den Upload der Datei mit dem angegebenen File-Transfer Protokoll.

Enthält <dateiname> keinen Pfad, wird die Datei aus dem Upload-Ordner bzw. Text-Upload-Ordner gelesen. Ist der Pfad relativ, basiert er auf dem Programmverzeichnis von ZOC oder auf dem Upload-Ordner.

Bei Transferprotokollen die die Übertragung mehrerer Dateien erlauben (Ymodem, Zmodem), kann der Parameter für den Dateinamen auch Namensmuster enthalten. Außerdem können mehrere Dateiname angegeben werden, indem die Namen durch das Pipe-Symbol (vertikaler Strich) getrennt werden: *.pdf|somefile.txt

Der Protokollname ist ASCII, BINARY oder einer der unter [Optionen→Verbindungsprofil→Dateitransfer](#) aufgeführten Protokolle, z.B. Zmodem oder Kermit.

Je nach Erfolg des Transfers, wird ##OK## oder ##ERROR## zurückgeliefert.

Beispiel:

```
CALL ZocUpload "ZMODEM", "ZOCBBS.REP"
sendet ZOCBBS.REP aus dem Upload-Ordner per Zmodem-Protokoll an die Gegenstelle.
```

Beispiel:

```
success= ZocUpload("XMODEM", "ZOC.ZIP")
sendet ZOC.ZIP (aus dem Uploadpfad) per Xmodem-Protokoll und fragt den Rückgabewert
(##OK## oder ##ERROR##) ab.
```

Beispiel:

```
CALL ZocUpload "ASCII", "commands.txt"
sendet die Datei commands.txt mit den momentan im Verbindungsprofil definierten Einstellungen
für den Text-Transfer.
```

Beispiel:

```
CALL ZocUpload "ASCII:CRONLY+10", "\FAR\AWAY\LIST.TXT"
```

sendet LIST.TXT per ASCII-Transfer mit Zeilenende nur-CR und einer ASCII-Verzögerung von 10 ms.

Beispiel:

```
CALL ZocUpload "ASCII:1+3", "HERE\SOME.DATA"
```

sendet die Datei SOME.DATA per ASCII-Transfer mit CR/LF-Übersetzung und einer ASCII-Verzögerung von 3ms.

Beispiel:

```
CALL ZocUpload "BINARY", "CNC-CONTROL.DAT"
```

überträgt den Inhalt der Datei ohne Umsetzung oder Übertragungsprotokoll.

Transfer Optionen

Mit Ausnahme des IND\$FILE Protokolls (siehe unten), werden die optionalen Einstellungen zur Konfiguration der Transferprotokolle durch eine Zeichenkette hinter dem Namen des Protokolls angegeben.

Passende Werte können erzeugt werden, indem die gewünschten Optionen unter [Optionen→Verbindungsprofil→Dateitransfer](#) eingestellt und mittels Shift+Strg+F10 in das Skript übernommen werden.

Beispiel: Es soll ein Xmodem-Transfer mit CRC und 1KB-Blöcken eingestellt werden. Hierzu gehen Sie folgendermaßen vor:

1. [Optionen→Verbindungsprofil→Dateitransfer](#) aufrufen, Xmodem wählen und diese Optionen aktivieren.
2. Das Verbindungsprofil-Fenster schließen (Speichern-Knopf)
3. Shift+Strg+F10 drücken
4. Der Parametertext wird im Status-Report als "Transfer-Options" ausgegeben, in diesem Fall "[0]kc".
5. Der REXX Befehl lautet dann `CALL ZocUpload "XMODEM:[0]kc", "datafile.zip"` (bei den Transfer-Optionen wird hierbei zwischen Groß- und Kleinschreibung unterschieden).

Transfer Optionen ASCII

Beim Dateitransfer im ASCII-Modus (gleichbedeutend mit Dateitransfer→Sende Textdatei) hat der Einstellungsparameter das Format "mode+chardelay+linedelay", wobei 'mode' als Nummer oder Text angibt, wie das Zeilenende konvertiert werden soll (ASIS= 0, CRLF= 1, CRONLY= 2, LFONLY= 3), und 'chardelay' die Sende-Verzögerung pro Zeichen angibt. Gültige Beispiele sind z.B. 2+5+100 oder CRONLY+5+100 (CR only mit 5 Millisekunden Sende-Verzögerung pro Zeichen und einer zusätzlichen Verzögerung von 100 ms am Ende jeder Zeile).

Transfer Optionen BINARY

Beim Dateitransfer im BINARY-Modus (gleichbedeutend mit Dateitransfer→Sende Binärdatei) kann der Einstellungsparameter eine Zahl sein, die die Verzögerung pro Zeichen in Millisekunden angibt (sonst wird die Verzögerung für Text-Senden aus dem Verbindungsprofil verwendet).

Transfer Optionen IND\$FILE

Was die Transfer-Optionen angeht, ist das IND\$FILE Protokoll eine Ausnahme. Der Optionen-Parameter für IND\$FILE enthält statt der oben beschriebenen Parameter das Kommando das am Host für den Transfer ausgeführt werden soll, z.B.:

```
CALL ZocUpload "IND$FILE:TSO IND$FILE PUT 'userid.projects.asm(report)'  
ASCII CRLF", "report.asm"
```

Siehe auch: [ZocDownload](#)

ZocWait(<text>)

Wartet auf den angegebenen Text. Bei Timeout-Überschreitung (siehe [ZocTimeout](#)) wird der Wert 640 zurückgeliefert.

Beispiel:

```
CALL ZocTimeout 20
timeout= ZocWait("Passwort")
IF timeout=640 THEN SAY "Keine Passwort-Aufforderung innerhalb 20 Sek"
ELSE CALL ZocSend "geheim^M"
```

Beispiel:

```
CALL ZocTimeout 10

timeout= ZocWait("Befehl eingeben>")
IF timeout=640 THEN SIGNAL the_end
CALL ZocSend "FIREWALL AKTIVIEREN^M"

timeout= ZocWait("Befehl eingeben>")
IF timeout=640 THEN SIGNAL the_end
CALL ZocSend "IPFILTER AKTIVIEREN^M"

timeout= ZocWait("Befehl eingeben>")
IF timeout=640 THEN SIGNAL the_end

SAY "Firewall und Ip-Filter wurden aktiviert!"

the_end:
Call ZocDisconnect
EXIT
```

Hinweis: Eventuelle Emulations-Steuercodes werden aus dem Datenstrom gefiltert und behindern den ZocWait-Befehl nicht (siehe [ZocWaitForSeq](#)).

Hinweis: Falls Sie ZocWait von einer DDE Applikation aus aufrufen muss dies via Request und nicht via Execute geschehen.

Hinweis: In Verbindung mit den TN3270 und TN5250 Emulationen kann nur auf das Zeichen ^Z gewartet werden, das in diesen Emulationen als Indikator dafür dient, dass das Terminal wieder eingabebereit ist.

Siehe auch: [ZocTimeout](#), [ZocWaitIdle](#), [ZocWaitLine](#), [ZocWaitMux](#), [ZocSyncTime](#)

ZocWaitForSeq 1|0|"on"|"off"

Normalerweise filtert ZOC SteuerCodes für die Terminalemulation aus dem Datenstrom, der für ZocWait verwendet wird. Mit diesem Befehl kann das Warten auf Steuersequenzen ein- oder

ausgeschaltet werden.

Dieser Schalter kontrolliert auch, ob [ZocReceiveBuf](#) solche SteuerCodes mitprotokolliert oder nicht.

Beispiel:

```
esc= ZocCtrlString("^[")  
  
/* wait for VT220 color reset */  
CALL ZocWaitForSeq "On"  
Call ZocWait esc||"[0m"
```

Siehe auch: [ZocWait](#), [ZocWaiMux](#), [ZocReceiveBuf](#)

ZocWaitIdle(<time>)

Dieser Befehl wartet auf eine Übertragungspause der angegebenen Länge, d.h. auf eine Zeit, in der die Gegenstelle keine Daten sendet. Falls keine solche Pause auftritt wird der Befehl nach der mit [ZocTimeout](#) gesetzten Zeit mit Returnwert 640 abgebrochen.

Beispiel:

```
CALL ZocTimeout 60  
timeout= ZocWaitIdle(2.5)  
IF timeout=640 THEN SAY "Gegenstelle sendete 60 Sek. ständig Daten"  
ELSE SAY "Endlich Ruhe! (2.5 Sekunden ohne Datenempfang)"
```

Siehe auch: [ZocWait](#), [ZocWaitLine](#), [ZocWaitIdle](#), [ZocWaitMux](#), [ZocWaitNumChars](#), [ZocTimeout](#), [ZocSyncTime](#)

ZocWaitLine()

Wartet auf die nächste nichtleere empfangene Zeile (zum Warten auf eine evtl. auch leere Zeile kann EXMLPL(ZocWait "^M") verwendet werden). Der zwischenzeitlich empfangene Text kann über die Funktion [ZocLastLine](#) bzw. über [ZocReceiveBuf](#) abgefragt werden.

Bei Timeout-Überschreitung liefert ZocWaitLine den Wert 640 zurück.

Hinweis: Da REXX Prozeduren in einem eigenen Thread laufen, ist es speziell in Schleifen möglich, dass Zeilen übersehen werden, wenn viel Text schnell empfangen wird (siehe auch ZocSynctime). Deshalb sollten solche Schleifen mit ZocWaitLine sehr kompakt gehalten sein und die Daten nur sammeln um sie dann erst nach dem Empfang verarbeiten (siehe zweites Beispiel).

Beispiel:

```
rc= ZocWaitLine()  
IF (rc\=640) & (ZocLastLine()="CONNECT") THEN ...
```

Beispiel:

```
/* Einen befehl schicken, der einige Zeilen an Output produziert */
CALL ZocSend 'dig emtec.com && echo "<<END>>"^M'

/* Nun zuerst die Daten komplett sammeln */
n= 0
DO FOREVER
    timeout= ZocWaitLine()

    /* Schleifenende wenn keine Daten empfangen */
    IF timeout=640 THEN LEAVE

    line= ZocLastLine()

    /* Zeile mit Endebedingung gefunden? */
    IF line=="<<END>>" THEN LEAVE

    /* Zeile nur in Array speichern, dann weiter mit nächster */
    n= n+1
    data.n= line
    data.0= n
END

/* Nun im 2ten Durchgang alle Zeilen im Array abarbeiten */
DO i= 1 TO data.0
    line= data.i

    /* 'line' kann hier in Ruhe verarbeitet werden
    ohne ggf. Daten zu verpassen */
END
```

Siehe auch: [ZocWait](#), [ZocWaiMux](#), [ZocWaitIdle](#), [ZocWaitNumChars](#), [ZocTimeout](#), [ZocSyncTime](#)

ZocWaitMux(<text0> [, <text1> ...])

Wartet auf einen von mehreren Texten in den Empfangsdaten. Die Bedingung ist erfüllt, wenn einer der Texte im Empfang gefunden wurde. Die Rückgabewert gibt Aufschluss darüber, welcher der Texte gefunden wurde (0, 1, 2, ...) bzw. ob ein Timeout aufgetreten ist (640).

Hinweis: Alle Texte zusammen dürfen eine Länge von 4096 Zeichen nicht überschreiten.

Beispiel:

```
CALL ZocTimeout 45
ret= ZocWaitMux("Mail eingetroffen", "Hauptmenü")
SELECT
    WHEN ret=0 THEN CALL handle_maildownload
    WHEN ret=1 THEN LEAVE
    OTHERWISE SIGNAL handle_error
END
```

Siehe auch: [ZocWait](#), [ZocWaitLine](#), [ZocWaitIdle](#), [ZocWaitNumChars](#), [ZocTimeout](#), [ZocSyncTime](#)

ZocWaitNumChars(<n>)

Wartet auf eine spezifische Anzahl von empfangenen Zeichen. Es handelt sich hierbei um jede Art von Zeichen, also auch Zeilenwechsel oder Steuerzeichen werden gezählt. Um abzufragen, welche Zeichen empfangen wurden, muss vor dem Aufruf von ZocWaitNumChars ein Empfangspuffer (siehe [ZocReceiveBuf](#)) eingerichtet, und dessen Inhalt ausgelesen werden, nachdem ZocWaitNumChars zurückgekehrt ist.

Wenn weniger als die angeforderte Anzahl von Zeichen innerhalb des Timeouts (siehe [ZocTimeout](#)) empfangen wurden, liefert die Funktion den Wert 640 zurück.

Beispiel:

```
CALL ZocTimeout 45
CALL ZocReceiveBuf 100
ret= ZocWaitNumChars(5)
IF ret\=640 THEN DO
    data= ZocReceiveBuf(0)
END
```

Siehe auch: [ZocReceiveBuf](#), [ZocWait](#), [ZocWaitIdle](#), [ZocWaitLine](#), [ZocTimeout](#), [ZocSyncTime](#)

ZocWindowState(MINIMIZE|MAXIMIZE|RESTORE|ACTIVATE|MOVE:x,y|QUERY)

Setzt den Status des Hauptfensters von ZOC auf den übergebenen Status (bzw. verschiebt das Fenster bei MOVE:x,y).

Als Funktion aufgerufen wird der neue Fensterstatus zurückgeliefert

Wird als Parameter **QUERY** übergeben, so wird nur der aktuelle Status des Fensters (**MINIMIZED**, **MAXIMIZED**, **RESTORED**) zurückgeliefert (bitte jeweils das **D** am Ende beachten).

Beispiel:

```
now= ZocWindowState("QUERY")
if now\="MINIMIZED" THEN DO
    CALL ZocWindowState "MINIMIZE"
end
```

ZocWrite <text>

Schreibt einen Text auf den Bildschirm. Dieser Befehl ist dem REXX-Befehl SAY sehr ähnlich, im Gegensatz zu **SAY** wird der Cursor aber nach der Textausgabe nicht auf die nächste Zeile gesetzt. Außerdem versteht ZocWrite [Steuerzeichen](#) wie **^M** (Enter) oder **^[** (ESC).

Beispiel:

```
CALL ZocWrite "Hello ^[[1m World^[[0m"
```

Siehe auch: [ZocSetScriptOutputDestination](#)

ZocWriteln <text>

Schreibt einen Text auf den Bildschirm und setzt den Cursor auf die nächste Zeile. Dieser Befehl ist ähnlich dem `SAY` Befehl, löst aber zusätzlich [Steuerzeichen](#) wie `^M` auf.

Beispiel:

```
CALL ZocWriteln "Hello ^M^J World"  
SAY "Hello" || X2C(0D) || X2C(0A) || "World"
```

Siehe auch: [ZocSetScriptOutputDestination](#)

Funktionen zur Kompatibilität mit älteren ZOC Versionen

ZocAutoConnect

Ersetzt durch [ZocConnectHostdirEntry](#).

ZocAskP

Umbenannt in [ZocAskPassword](#).

ZocGetFilename(s)

Umbenannt in [ZocAskFilename](#) und [ZocAskFilesnames](#).

ZocGetFolderName

Umbenannt in [ZocAskFolderName](#).

ZocBaud

Ersetzt durch [ZocSetDeviceOpts](#), z.B.: `Call ZocSetDeviceOpts "[1]38400-8N1"`.

ZocCaptClr

Ersetzt durch `ZocCommand("CLEARSCROLLBACK")`.

ZocCarrier

Ersetzt durch [ZocGetInfo\("ONLINE"\)](#).

ZocCls

Umbenannt in [ZocClearScreen](#).

ZocCursor

Ersetzt durch [ZocGetInfo\("CURSOR-X"\)](#) und [ZocGetInfo\("CURSOR-Y"\)](#).

ZocDial()

Umbenannt in [ZocConnect](#).

ZocEndZoc()

Ersetzt durch [ZocTerminate](#).

ZocExec

Umbenannt in [ZocShellExec](#).

ZocGetLine

Umbenannt in [ZocWaitLine](#).

ZocGetOption

Umbenannt in [ZocGetSessionOption](#).

ZocGetPhonebk

Umbenannt in [ZocGetHostEntry](#).

ZocGlobal

Umbenannt in [ZocGlobalStore](#).

ZocLockKeyboard

Ersetzt durch ZocKeyboard("LOCK") und ZocKeyboard("UNLOCK").

ZocLoadOpts

Umbenannt in [ZocLoadSessionProfile](#).

ZocLoadKeyfile

Umbenannt in [ZocLoadKeyboardProfile](#).

ZocLogging

Umbenannt in [ZocSetLogging](#).

ZocLogname

Umbenannt in [ZocSetLogfileName](#).

ZocRestimer

Ersetzt durch ZocSetTimer("00:00:00").

ZocSaveOpts

Ersetzt durch [ZocSaveSessionProfile](#).

ZocScreen

Ersetzt durch [ZocGetScreen](#).

ZocSendBreak

Ersetzt durch ZocCommand("SEENDBREAK").

ZocSendKey <nummer>

Bitte verwenden Sie stattdessen [ZocGetSessionOption](#) und [ZocSend](#).

ZocSetDevParm

Umbenannt in [ZocSetDeviceOpts](#).

ZocSetDIPath

Bitte verwenden Sie stattdessen `CALL ZocSetProgramOption "DownloadPath=<path>"`.

ZocSetEmu

Bitte verwenden Sie stattdessen `CALL ZocSetSessionOption "ActiveEmulation=<emu-id>"` oder ZocSetEmulation.

ZocSetHost

Bitte verwenden Sie stattdessen `CALL ZocSetSessionOption "Host=yes|no"`.

ZocSetPhonebk

Umbenannt in [ZocSetHostEntry](#).

ZocSetOption

Umbenannt in [ZocSetSessionOption](#).

1.11.11 ZOC-REXX ZocDialog-Vorlagen

Eine Vorlage für den Befehl [ZocDialog](#) bezeichnet eine Textdatei, welche eine Liste und Beschreibung aller Elemente, die im Dialogfenster erscheinen sollen, enthält.

Das grundsätzliche Format eines Dialog-Vorlagen Eintrages ist

```
<typ> <element-id> = "<text>", <wert1>, <wert2>[, <wert3>[, <wert4>]]
```

z.B.

```
DIALOG IDDLG0 = "Bitte Benutzernamen eingeben", 260,260
```

Der *typ* eines Eintrages wird durch das erste Wort dieses Eintrags in Großschreibung bestimmt. Die *element-id* ist ein vom Benutzer definierter Text, der dazu verwendet werden kann, nach Schließen des Dialogs den Wert des entsprechenden Elements abzufragen (z.B. um dann die Eingabe des Benutzers aus einem Textfeld auszulesen). Der Wert von *text* hängt vom Typ des Elements ab und enthält üblicherweise entweder die Bezeichnung oder den Inhalt des Elements. Abhängig vom Typ des Elements geben *wert1/wert2* entweder die x/y-Koordinate innerhalb des Fensters oder dessen Größe an. Die Werte *wert3/wert4* bestimmen die Größe (Breite/Höhe) derjenigen Elemente, für die *wert1/wert2* zur Angabe der Position verwendet wird. Wird für Breite und Höhe kein Wert angegeben, erhält das Element eine Standardgröße.

Der Interpreter für obige Befehle ist sehr einfach gehalten und nicht fehlertolerant. Bitte treffen Sie keine Annahmen über mögliche andere Schreibweisen, die über die hier beschriebenen Beispiele hinausgehen, und nehmen Sie sich keine syntaktischen Freiheiten. Nur ein Dialog-Element pro Zeile ist zulässig. Zeilen, die mit *//* oder *--* oder *#* beginnen, werden als Kommentarzeile behandelt. Leerzeilen sind erlaubt.

Liste der Möglichen Einträge in Dialogvorlagen:

DIALOG

```
DIALOG <element-id> = "<titel>", <breite>, <höhe>
```

Bestimmt den Fenstertitel und die Größe des Dialogs.

STATICTEXT

```
STATICTEXT <element-id> = "<text>", <x-pos>, <y-pos>[, <breite>[, <höhe>]]
```

Erstellt ein beschreibendes Text-Element (z.B. ein Label).

EDITTEXT

```
EDITTEXT <element-id> = "<inhalt>", <x-pos>, <y-pos>[, <breite>[, <höhe>]]
```

Erstellt ein Eingabefeld für Text, vorbelegt mit dem angegebenen Inhalt. Die *element-id* kann dazu verwendet werden, den Inhalt des Textfeldes auszulesen, nachdem der Dialog verarbeitet wurde.

EDITPASSWORD

Wie [EDITTEXT](#), jedoch wird die Eingabe nicht angezeigt.

CHECKBOX

```
CHECKBOX <element-id> = "<titel>", <x-pos>, <y-pos>[, <breite>[, <höhe>]]
```

Erstellt eine Checkbox mit entsprechender Überschrift an der angegebenen Position. Die *element-id* kann dazu verwendet werden, den Zustand der Checkbox abzufragen, nachdem der Dialog verarbeitet wurde. Wenn der Text mit *[*]* beginnt, wird das Element aktiviert angezeigt.

RADIOBUTTON

```
RADIOBUTTON <element-id> = "<titel>", <x-pos>, <y-pos>[, <breite>[, <höhe>]]
```

Erstellt einen Radiobutton mit entsprechender Überschrift an der angegebenen Position. Die *element-id* kann dazu verwendet werden, den Zustand des Radiobuttons abzufragen, nachdem der

Dialog verarbeitet wurde. Wenn der Text mit [*] beginnt, wird das Element aktiviert angezeigt.

GROUPBOX

GROUPBOX <element-id> = "<titel>", <x-pos>, <y-pos>, <breite>, <höhe>

Erstellt einen rechteckigen Rahmen innerhalb des Fensters um Inhalt optisch zu gruppieren oder herauszustellen. Es sind maximal fünf Gruppierungen pro Dialog möglich.

DROPDOWNLIST

DROPDOWNLIST <element-id> = "<inhalt>", <x-pos>, <y-pos>, <breite>, <höhe>

Erstellt ein Mehrfachauswahl-Feld mit einer Liste von Einträgen bei dem die Liste heruntergeklappt werden kann. Die einzelnen Einträge werden über das Inhalt-Feld, getrennt mit je einem vertikalem Strich, angegeben (z.B. "Ja|Nein|Vielleicht"). Wenn ein Eintrag mit [*] beginnt, wird dieser Eintrag vorselektiert. Die element-id kann dazu verwendet werden, um den Namen des ausgewählten Eintrags abzufragen, nachdem der Dialog verarbeitet wurde. Es sind maximal fünf Mehrfachauswahl-Felder pro Dialog möglich.

LISTBOX

LISTBOX <element-id> = "<inhalt>", <x-pos>, <y-pos>, <breite>, <höhe>

Erstellt ein Mehrfachauswahl-Feld mit einer Liste von Einträgen bei der alle Einträge untereinander dargestellt werden. Die Werte für die einzelnen Einträge werden über das Inhalt-Feld, getrennt mit je einem vertikalem Strich, angegeben (z.B. "Ja|Nein|Vielleicht"). Wenn ein Eintrag mit [*] beginnt, wird dieser Eintrag vorselektiert. Die element-id kann dazu verwendet werden, um den Namen des ausgewählten Eintrags abzufragen, nachdem der Dialog verarbeitet wurde.

PUSHBUTTON

PUSHBUTTON <element-id> = "<text>", <x-pos>, <y-pos>[, <breite>[, <höhe>]]

Erstellt eine Schaltfläche, z.B. zum Schließen des Dialogs bei Betätigung. Die Schaltfläche, die zum Schließen des Dialogs verwendet wurde, liefert bei Abfrage mit ZocDialog GET und der element-id den Wert "1" zurück.

DEFPUSHBUTTON

Identisch mit PUSHBUTTON, erstellt aber eine Schaltfläche, die als Standard-Schaltfläche markiert ist (DEFPUSHBUTTON wird üblicherweise für OK- oder Fertig-Schaltflächen verwendet).

Beispiel: test.dlg

```
// Nachfolgend die erste Dialog-Vorlage innerhalb der Vorlage-Datei test.dlg
DIALOG MSG1 = "Einfacher Beispiel-Dialog", 260,140
STATICTEXT ST = "Glückwunsch! Dies ist Ihr erster Dialog.", 12,12, 100,60
DEFPUSHBUTTON P1 = "OK", 120,70
```

```
// Hier eine weitere Dialog-Vorlage innerhalb der gleichen Vorlage-Datei
DIALOG MAIN = "Beispiel-Dialog", 260,310
STATICTEXT ST1 = "Ihr Name", 12,12, 100
EDITTEXT ED1 = "Harry", 112,10
CHECKBOX CB1 = "Ich bin Programmierer", 14,50, 150
GROUPBOX G1 = "Lieblingsfarbe?", 12, 90, 200,90
DROPDOWNLIST DD1= "Rot|[*]Gruen|Blau", 20,115, 170,100
RADIOBUTTON RB1 = "VT220", 14,190, 80
RADIOBUTTON RB2 = "[*]Xterm", 100,190, 80
DEFPUSHBUTTON P1 = "OK", 12, 230, 80
PUSHBUTTON P2 = "Abbruch", 112, 230, 80
```

Beispiel: Verwendung der obigen Vorlage innerhalb eines Scripts

```
-- Anzeigen einer einfachen Nachricht
CALL ZocDialog "SHOW", "MSG1@test.dlg"

-- Anzeiges eines Dialogs und Prüfung der eingegebenen Werte
dlgrc= ZocDialog("SHOW", "MAIN@test.dlg")
SAY ">"dlgrc
IF dlgrc=="##OK##" THEN DO
    okpressed= ZocDialog("GET", "P1")
    IF okpressed==1 THEN DO
        SAY "EDITTEXT Wert: "||ZocDialog("GET", "ED1")
        SAY "CHECKBOX Wert: "||ZocDialog("GET", "CB1")
        SAY "DROPDOWNLIST Wert: "||ZocDialog("GET", "DD1")
    END
END
END
```

1.11.12 Leicht zu übersehende Funktionen

- ☐ Wenn die Maus längere Zeit über einer der LEDs in der Statuszeile steht, erscheint eine kurze Beschreibung dieser LED. Das gleiche gilt für Symbolleiste und Benutzerknöpfe.
- ☐ Ein rechter Mausklick in die Symbolleiste öffnet das Settings-Fenster der Symbolleiste.
- ☐ Ein rechter Mausklick in die Benutzerknöpfe öffnet das Einstellungsfenster für die Benutzerknöpfe.
- ☐ Ein rechter Mausklick in die Statuszeile öffnet ein Popup-Menü mit allen Pulldown-Menüpunkten.
- ☐ Ein rechter Mausklick in das ZOC-Hauptfenster öffnet ein Popup-Menü.
- ☐ Mit gedrückter Alt-Taste können Rechtecke am Bildschirm markiert werden.
- ☐ Das Markieren einer Textzeile mit der Maus bei gleichzeitigem Drücken der Shift-Taste sendet diesen Text sofort.
- ☐ Die Funktion 'Text zitieren' formatiert den Text bei gleichzeitigem Drücken der Strg-Taste automatisch auf Bildschirmbreite um.
- ☐ Ein rechter Mausklick ins Adressbuch öffnet ein Popup-Menü.
- ☐ Die Option Menüzugriff unter Adressbucheintrag ändern setzt den jeweiligen Adressbucheintrag ins Datei-Menü.
- ☐ Der Minimieren/Maximieren Knopf im AutoWahl-Fenster schaltet die Größe des Fensters zwischen groß und klein um (kann auch in den Adressbuch-Optionen eingestellt werden).
- ☐ Das Schnipsel-Fenster (Optionen, Fenster) speichert nützliche Informationen wie Internet-Adressen usw.
- ☐ Die Eingabezeile (Alt+C) besitzt eine History-Funktion (Cursor up/down).
- ☐ Der Dialog zur Tastatur-Umbelegung erlaubt verschiedene Tastenbelegungen je nach Zustand der Rollen-Taste. Auf diese Weise können zwei völlig unterschiedliche Tastaturbelegungen mit nur einer Taste umgeschaltet werden.

1.11.13 'Wie geht das?'-Fragen

Bitte lesen Sie auch die [Einstiegsanleitung](#) für Hinweise wie Verbindungen per Telnet, SSH, ISDN usw. aufgebaut werden und [Häufige Probleme](#).

Wie wird eine Verbindung per serielllem Kabel, Telnet, SSH oder ISDN aufgebaut?

Bitte lesen Sie unsere [Einstiegsanleitung](#).

Wie kann ich ZOC Einstellungen von einem Rechner auf einen anderen kopieren?

Durch die Funktion [Dateimenü→Datenordner öffnen](#) wird ein Fenster mit dem Ordner angezeigt, der alle Einstellungen enthält.

Die Dateien aus diesem Ordner (und speziell dem Unterordner [Options](#)) können Sie auf einen USB-Stick o.Ä. transferieren.

Auf dem neuen Rechner starten Sie ZOC, wählen [Dateimenü→Datenordner öffnen](#), beenden ZOC und Kopieren die Dateien vom USB-Stick dorthin.

Wie kann ich den ZOC Datenordner an einer anderen Stelle speichern?

Verschieben Sie den Ordner ([Dateimenü→Datenordner öffnen](#)) einfach im mit Hilfe von Funktionen des Betriebssystems. Wenn ZOC beim nächsten Start den Ordner nicht mehr findet, fragt das Programm nach der neuen Position des Ordners.

Wie kann ich die Sprache zwischen deutsch und englisch umstellen?

Fügen Sie `/LANG:ENGL` oder `/LANG:GER` zur Kommandozeile hinzu (entweder im ZOC Icon oder durch eine Commandline.ini Datei). Siehe [ZOC Befehlszeile](#).

Wie werden Schriftarten und Zeichensätze geändert

Die Konfiguration hierfür geschieht unter ([Optionen→Verbindungsprofil→Farben](#) und [Optionen→Verbindungsprofil→Layout](#)). Bitte lesen Sie hierzu auch das Thema [Schriftarten, Fenstergröße, Farben](#).

Wie kann man eine SSH Schlüsseldatei erzeugen?

Es gibt dazu zwei Möglichkeiten. Machen Sie Secure Shell zur aktiven Verbindungsmethode (Optionen-Menü→Verbindungsprofil). Dann finden Sie im Datei-Menü einen Menüpunkt zum Erzeugen von Schlüsseln. Alternativ können Sie über die Windows Eingabeaufforderung das Tool `ssh-keygen.exe` benutzen, das Sie im ZOC Programmordner finden (das Tool funktioniert so wie das gleichnamige OpenSSH `ssh-keygen`, hierfür sind im Internet viele Beispiele auffindbar).

Wie kann ich das Einfügen aus der Zwischenablage beschleunigen?

Diese Option ist in [Optionen→Verbindungsprofil→Transfer→Text-Senden](#).

Wie kann Dateien auf einen Linux-Rechner übertragen

Sie können hierfür Zmodem oder SCP verwenden. Bitte rufen Sie hierzu das Hilfethema ganz unten im Transfermenü auf: [Transfer Überblick](#)

Wie kann ich ein Icon auf der Arbeitsoberfläche für eine Verbindung erstellen?

Im Dialog [Datei→Neue Verbindung](#) gibt es hierfür die Funktion [Speichern unter](#). Falls Sie für die Verbindung bereits einen Eintrag im Adressbuch haben, können Sie dort im Eintrag im Register [Verknüpfung](#) ein Icon erstellen.

Gibt es noch andere Schriftarten für ZOC?

ZOC durchsucht die Systemschriftarten und nutzt diejenigen, bei denen die Zeichen alle die gleiche Breite haben. Wenn Sie solche Schriftarten zum System hinzufügen (z.B. per Googlesuche nach "Inconsolata" oder "ProFont") werden sie nach Neustart von ZOC angeboten.

Wie kann ich die Warnmeldungen ausschalten?

Die Warnmeldungen können in [Optionen→Programmeinstellungen→Abfragen](#) deaktiviert werden.

Wie kann ich gespeicherte SSH Kennwörter löschen

Die Kennwortverwaltung ist in [Optionen→Programmeinstellungen→Kennwörter](#), Kennwörter.

Wie bediene ich Cisco Router die am seriellen Port angeschlossen sind?

Siehe [Einstieg für serielle Verbindungen](#).

Wie installiere ich ZOC in einem LAN?

Siehe [Netzwerk](#) und [System/Netzwerk Administratoren](#).

Wie kann ich mehr als eine Seite Text im Anzeigepuffer-Fenster markieren?

Entweder Sie markieren die Anfangsstelle im Text, scrollen zur nächsten Seite und markieren mit gedrückter Strg-Taste die Endstelle, oder Sie verwenden Log, Puffer editieren anstelle des Anzeigepuffer-Fensters von ZOC.

Wie kann ich ein Break-Signal an einen Cisco Router schicken?

Wenn eine Verbindung per Telnet oder Seriell aktiv ist, erscheint dafür ein Eintrag im Datei-Menü von ZOC.

Wie installiere ich ZOC als Default-Telnet für Internet Explorer/Firefox/Safari?

Windows: Öffnen Sie den Windows Explorer (nicht Internet Explorer), gehen sie zu Anzeige, Optionen, Dateitypen, URL-Telnet, Bearbeiten, Öffnen, Bearbeiten und geben Sie

```
"C:\Programme\Zoc\Zoc.exe" %1
```

macOS: Unter Firefox können Sie das Programm für SSH: oder TELNET: Links wählen, wenn Sie einen solchen Link anklicken. In diesem Fall wählen Sie einfach ZOC aus.

Um das Standardprogramm für derartige Links in Safari oder dem macOS Finder zu ändern brauchen Sie eine Software die die URL-Handler für macOS bearbeiten kann. Solche Programme sind zum Beispiel [More Internet](#) oder [RCDefaultApp](#).

Wie kann ich ZOC auf Tonwahl umstellen?

In den Einstellungen für den Verbindungstyp Serial/Modem können die AT-Befehle festgelegt werden. Dort ggf. einen Wählcode auf 'ATDT' umstellen.

1.11.14 Häufige Probleme, Fragen und Antworten

Bitte lesen Sie auch die [Einstiegsanleitung](#) für Hinweise wie Verbindungen per Telnet, SSH, ISDN usw. aufgebaut werden. Außerdem bieten auch die ['Wie geht das?'-Fragen](#) mögliche Lösungen.

Falls Sie auch weiter unten keine Lösung finden, können Sie über unsere Webseite im Supportbereich von uns eine Antwort bekommen.

ZOC versucht beim Starten auf dem COM Port zuzugreifen, obwohl keine serielle Verbindung gewünscht wird

Evtl. ist im Standard-Verbindungsprofil [Serial/Modem](#) als Verbindungstyp gewählt (was dann beim Laden aktiviert ist). Bitte stellen Sie nach dem Start in [Optionen→Verbindungsprofil→Verbindungstyp](#) den Standard-Verbindungstyp auf Telnet oder SSH um und speichern Sie das Profil Standard.zoc wieder ab.

Bei Telnet-Verbindungen zu IP-Adressen entsteht eine längere Pause bevor die Verbindung aktiv wird

Evtl. findet ZOC für die IP Adresse keinen zugehörigen Namen über den Namensserver (DNS). Gehen Sie zu [Optionen→Verbindungsprofil→Verbindungstyp→Telnet→Erweiterte Optionen](#) und schalten Sie 'Kein Hostname zu IP Adresse suchen' ein.

Wenn die Fenstertransparenz aktiviert ist, wird der Rechner deutlich langsamer

Um die Transparenzfunktion sinnvoll zu nutzen, benötigen Sie eine leistungsfähige Graphikkarte, da anderenfalls der Systemprozessor die Arbeit hierfür übernehmen muss. Mit älteren Karten sollte die Option in den [Programmeinstellungen](#) deaktiviert werden.

ZOC zeigt Umlaute oder Sonderzeichen nicht korrekt an

Bitte versuchen Sie eine andere Emulation zu verwenden (beste Chancen sind in der Regel [VT220](#) oder [Xterm](#)). Aktivieren Sie alternativ unter [Verbindungsprofil→Layout](#) auch [Unicode](#) oder experimentieren Sie dort mit den anderen Zeichensätzen (z.B. [IBM/DOS](#), [Windows/Latin](#), etc.). Falls dies nicht hilft, wenden Sie sich bitte an unseren Support.

Auf Linux Systemen werden bei Yast/MC/etc. seltsame Zeichen angezeigt

Um Rahmenzeichen, Sonderzeichen usw. korrekt anzuzeigen, muss für die meisten Linux Systeme in ZOC in den Einstellungen der verwendeten Emulation ([Xterm/VT220/Linux](#)) oder unter [Verbindungsprofil→Layout](#) der Zeichensatz [UTF8/Unicode](#) ausgewählt werden.

Die Emulation funktioniert im Prinzip, aber Linien und Kästen werden falsch dargestellt

Gehen Sie zu [Session Profile, Layout](#) und probieren Sie beim Zeichensatz entweder [UTF8/Unicode](#) oder [IBM/DOS](#).

Nicht alle F-Tasten funktionieren mit meiner Gegenstelle

Es ist wichtig, dass Sie eine Emulation verwenden, die die volle Anzahl F-Tasten unterstützt. [VT220](#), [Sun-CDE](#), [Ansi-SCO](#) sind hierbei gute Möglichkeiten (für [VT220](#) gibt es zwei F-Tastenstandards, die sie in den Emulationsoptionen umstellen können). Ansonsten, falls Sie wissen, welche Codes die Gegenstelle für die F-Tasten erwartet, können Sie dies per [Optionen→Verbindungsprofil→F-Tasten](#) oder per [Optionen→Tastaturprofile](#) festlegen.

Die Cursortasten funktionieren nicht in allen VT102/VT220 Programmen

Bitte benutzen Sie die grauen Cursortasten auf Ihrer Tastatur (nicht den Nummernblock). Der Nummernblock wird von manchen Programmen für andere Zwecke benutzt.

Die globale Tastaturbelegung funktioniert nicht

Bitte beachten Sie, dass diese Art der Umdefinition den Status der Num-Taste beachtet. Prüfen Sie bei den Tastatenprofilen, ob die Taste in Abhängigkeit vom Num-Lock Status definiert ist.

Bei jedem Verlassen des Programms erscheint eine Carrier Detect Warnung?

Eventuell ist das Modem auf 'CD always on' eingestellt. Entweder muss das Modem mit [AT&C1](#) konfiguriert oder die CD Signal-Erkennung im Menü [Optionen→Verbindungsprofil→Verbindungstyp→Serial/Modem](#) (CD Prüfung) abgeschaltet werden.

Die Größe des ZOC-Fensters kann nicht stufenlos eingestellt werden

Normalerweise ändert ZOC die Größe der Schriftart mit der Größe des Fensters. Da nicht jede Schriftart in jeder Größe vorhanden ist, verwendet ZOC die nächstliegende Schriftart und passt die Größe des Fensters dieser Schriftart an. Die Art, wie ZOC sich einer Größenveränderung anpasst, kann in [Optionen→Verbindungsprofil→Layout](#) eingestellt werden.

Ich möchte in ZOC zusätzliche Schriftarten benutzen können

ZOC überprüft beim Programmstart alle verfügbaren Schriftarten auf die Verwendbarkeit in einem Terminalfenster. Für eine Verwendung in ZOC müssen alle Zeichen einer Schriftart die gleiche Breite besitzen und die Schriftart muss die in ZOC eingestellte Code-Seite (Codepage, Zeichensatz, z.B. [Windows/ANSI](#) oder [IBM/OEM](#)) unterstützen.

Unter Windows treffen diese Bedingungen auf die Zeichensätze [Courier New](#), [Lucida Console](#) und [Terminal](#) zu. Sollte die Schriftart [Terminal](#) in ZOC's Fontliste nicht erscheinen, stellen Sie bitte sicher, dass unter [Arbeitsplatz](#), [Systemsteuerung](#), [Schriftarten](#), [Ansicht](#), [Optionen](#), [True Type](#) die Einstellung 'Nur True Type Schriftarten verwenden' nicht aktiviert ist.

Ich habe das Menü deaktiviert, wie bekomme ich es zurück

Sie können den Terminalbereich mit der rechten Maustaste anklicken. Falls Sie die rechte Maustaste mit einer anderen Funktion belegt haben, können Sie dies über die

Programmeinstellungen (Tastenkombination Alt+Komma) rückgängig machen. Alternativ können Sie die Datei mit dem Verbindungsprofil (z.B. Eigene Dateien → ZOC9 Dateien → Options → Standard.zoc) mit einem Text-Editor direkt bearbeiten und die Option zurücksetzen.

Ich vermisse einen Vollbildmodus

Es gibt keinen zeichenorientierten Vollbildmodus, da ZOC eine echte Fensteranwendung ist. Mit dem Maximierungs-Knopf kann ZOC aber auf die gesamte Bildschirmgröße vergrößert werden. Bei Verwendung einer großen Schriftart füllt ZOC dann den gesamten Bildschirm.

ZOC scrollt zu langsam

Scrollen in grafischen Fenstern ist generell langsamer als im Vollbild-Modus. Allerdings gibt es in ZOC eine Option, um das Scrollen zu beschleunigen (siehe [Optionen→Verbindungsprofil→Fenster](#)).

ZOC kann den COM-Port nicht öffnen, andere Programme schon

Bitte testen Sie, ob der COM-Port verfügbar ist, indem Sie in der Eingabeaufforderung MODE COMx (x ist die Nummer des COM-Ports) eingeben.

Erscheint keine Fehlermeldung, überprüfen Sie, ob die Bezeichnung des COM-Port in den ZOC-Einstellungen richtig geschrieben ist (ist vielleicht ein Leerzeichen zwischen COM und der Port-Nummer vorhanden). Manche Systeme benötigen auch die Schreibweise `\\.\COMxx`

Erscheint eine Fehlermeldung nach MODE COMx, überprüfen Sie bitte die Ports in der Systemsteuerung bzw. ob ein anderes Programm den Port belegt.

Ich bekomme die Meldung "Kein CTS vom Modem"

Um zu verhindern, dass ZOC und das Modem sich gegenseitig den internen Puffer überschreiben, sollten beide so eingestellt sein, dass die RTS/CTS Steuerung unterstützt wird. Dies ist insbesondere bei Übertragungs-Geschwindigkeiten größer 9600 bps notwendig. Wie andere Terminal-Programme besteht auch ZOC bei diesen Geschwindigkeiten auf das RTS/CTS Signal. Deshalb gibt ZOC eine Meldung aus, wenn vom Modem kein CTS-Signal erkannt wird.

Wenn diese Meldung erscheint, überprüfen Sie bitte Ihre Modem-Konfiguration auf CTS-Unterstützung (stellen Sie ggf. Ihr Modem auf die werkseitige Einstellung), verwenden Sie ein 7-adriges Verbindungskabel zwischen Modem und Computer und überprüfen Sie, ob der korrekte COM-Port unter [Optionen→Verbindungsprofil→Verbindungstyp](#) bzw. unter [Adressbuch→Ändern](#) eingestellt ist.

Mein Modem funktioniert mit anderen Programmen, aber nicht mit ZOC

Das Modem muss genauso initialisiert werden wie in anderen DFÜ-Programmen. Alternativ kann die Modemkonfiguration auch im Modem abgespeichert werden (meist mit AT&W). Danach kann ZOC das Modem mit ATZ auf diese Konfiguration zurückstellen.

Anwahlversuche mit Windows-Modems schlagen fehl

Vielleicht fehlt eine 0 am Anfang der Nummer, um eine Amtsleitung zu bekommen. Der Verbindungstyp Windows-Modem gibt die angegebene Nummer direkt weiter, ohne auf eine eventuelle Einstellung in den Standorteinstellungen zu achten.

Nach dem Anwählen einer Mailbox erscheinen keine Zeichen am Bildschirm

Das Modem wartet auf ein RTS/CTS Signal des Computers. In den seriellen Einstellungen muss RTS/CTS eingeschaltet werden.

Downloads funktionieren, aber bei großen Uploads mit Modem treten Fehler auf

Mit hoher Wahrscheinlichkeit ist eine hohe Übertragungsgeschwindigkeit ohne RTS/CTS die Ursache. Für schnelle Übertragungen ist es notwendig, bei ZOC und Modem das RTS/CTS Protokoll einzuschalten.

Dateiübertragungen mit Zmodem funktionieren nicht unter Telnet

Bitte probieren Sie unterschiedliche Telnet-Optionen (speziell z.B. die Option CR/NUL). Alternativ starten Sie das entfernte Zmodem mit dem Parameter `-e`.

Zmodem Uploads zum Unix-Programm RZ starten nicht oder enden mit Fehlern

Versuchen Sie auf der Unix Seite die -e, -eb oder -v Option: `rz -e`, `rz -eb` oder `rz -v`.

Wenn die Verbindung durch einen Xyplex-Konzentrator geroutet wird, ist uns momentan keine Methode bekannt, um Zmodem Uploads zu machen. In diesem Fall muss auf ein anderes Protokoll zurückgegriffen werden. Die beste Möglichkeit ist in diesem Fall wahrscheinlich ein aus der ZOC Transfer-Shell gestartetes CKERMIT (für genauere Infos durchsuchen Sie den Hilfetext bitte nach 'CKERMIT').

Gibt es einen Hostmodus?

Ja. Der Hostmodus ist als REXX-Programm implementiert. Zum Starten muss das Programm MINIHOST.ZRX unter Skript, REXX starten ausgewählt werden. Das Programm fragt zwei Passwörter ab (für Gäste und Berechtigte) und wartet dann auf eingehende Anrufe.

Das Datum von übertragenen Dateien ist um einige Stunden verschoben

Bitte lesen Sie den Hilfetext zur SET-Variable TZ im Kapitel 'ZOC starten' oder setzen Sie die Option 'Datum/Zeit lokal' in [Einstellungen Transfer](#).

Wenn am Ende eines File-Transfers ein Menü offen ist, hängt ZOC

Dies ist ein bekanntes Problem, gegen das momentan nichts unternommen werden kann. Bitte vermeiden Sie diese Situation.

ZOC zeigt unter Windows den Verbindungstyp ISDN nicht an

Sie brauchen dazu CAPI V2.0 Treiber Ihres Herstellers, die mit praktisch allen ISDN-Karten mitgeliefert werden.

ISDN lässt sich einstellen, aber Verbindungen schlagen fehl

Es wird entweder eine CAPI2032.DLL verwendet, die nicht zum aktuell geladenen Treibersatz passt, oder es sind in ZOC falsche ISDN-Parameter eingestellt. Bitte verwenden Sie zum Einstellen der Parameter den Knopf STANDARD unter [Optionen→Verbindungsprofil→Verbindungstyp→ISDN](#) und speichern Sie dann das Verbindungsprofil ab. Beachten Sie auch, dass jeder einzelne Adressbucheintrag eigene ISDN-Parameter besitzt und die Standardeinstellung auch hier eingestellt und abgespeichert werden müssen.

Sollten trotzdem Fehler auftreten, stellen Sie bitte sicher, dass Sie die aktuellste Treiberversion des jeweiligen Kartenherstellers verwenden.

Was bedeutet ZOC?

Der 'Hitchhikers Guide to the Galaxy' könnte ZOC wie folgt beschreiben:

Zoc (n), *Zoc* bedeutet Kommunikation mit Hilfe von extrem hochentwickelter Hard- und Software (hochentwickelt zumindest aus Sicht der vom Affen abstammenden Lebensform, die einen völlig unwichtigen blauen Planeten in einem noch unwichtigeren Seitenarm der Milchstraße bewohnt, und die glaubt, dass Personal Computer eine wirklich tolle Sache sind).