

# Euroscope Befehle und Alias

## Prolog

Euroscope und die implementierten PlugIn bieten die Möglichkeit über Befehle bestimmte Funktionen zu steuern oder Aktionen auszuführen. Mit einem Alias wird dem Controller die Möglichkeit gegeben, Standardphrasen (gerade bei Textpiloten) schnell zu erstellen und zu versenden, ohne jedes Mal den kompletten Text von Hand einzugeben.

## Handhabung

Jeder Befehl / Alias beginnt erstmal mit einem "." [Punkt] und wird in die Euroscope Befehlsleiste eingegeben. Wir müssen zwischen drei verschiedenen Verfahrensweisen unterscheiden:

### 1. Eigenständige Befehle

Diese beginnen mit einem Punkt, gefolgt von dem Befehl, eventuellen Parametern und werden mit der ENTER-Taste ausgelöst. Dabei ist es unerheblich ob ein Tag / Label ausgewählt ist.

*Beispiel:* `.delhel auto`

### 2. Befehle, die mit einem Flieger in Verbindung stehen

Dazu wird zuerst der Befehl eingegeben und danach das Tag / Label angeklickt.

*Beispiel:* `.chat` [Klick auf Tag / Label]

Eine Alternative wäre: `.chat [Callsign]`, also `.chat TUI1RG`

### 3. Alias, um Textnachrichten an den Flieger zu senden

Hierbei ist es erforderlich zuerst das Tag / Label auszuwählen (Kontrolle Callsign in der Status- oder Kopfzeile von Euroscope) und danach die Textnachricht über den Alias vorzubereiten und zu versenden. Um dies zu realisieren sind mindestens zwei Schritte erforderlich.

*Beispiel 1:*

`.rbc` [Leertaste] der Befehl wird auf „readback correct“ aufgelöst und kann nun mit der ENTER-Taste verschickt werden.

*Beispiel 2:*

`.co` [Leertaste] der Befehl wird auf „Contact \$radioname(\$1) on frequency \$freq(\$1), good bye!“ aufgelöst. Der Fokus ist bereits auf das „\$1“ gesetzt. Hier wird einfach das Stationskürzel aus der Controllerlist eingesetzt, z. B. „man“.

Wenn ihr jetzt die TABULATOR-Taste drückt, erfolgt die Auflösung auf „Contact Muenchen Radar on frequency 123.900, good bye!“ Nun kann der Text mit Druck auf die ENTER-Taste an den zuvor selektierten Flieger verschickt werden.

# Übersicht über die wichtigsten Befehle

## 1. Eigenständige Befehle

Befehl	Erklärung
<code>.delhel auto</code>	Startet den DeliveryHelper um die Departurelist zu pflegen (an allen aktiv gestellten Airports). Beendet den DeliveryHelper um die Departurelist zu pflegen (an allen aktiv gestellten Airports).
<code>.cdm master EDMM</code>	Die eingeloggte Station wird als Master für das A-CDM-Tool definiert.
<code>.cdm slave</code>	Die eingeloggte Station gibt die Masterrolle wieder ab und wird zum slave.

## 2. Befehle für Flieger

Befehl	Erklärung
<code>.chat</code> [Klick auf Tag / Label] oder F1 + C [Klick auf Tag / Label]	Startet eine neue Chatnachricht mit dem angeklickten Callsign
<code>.am</code> [Klick auf Tag / Label] oder F1 + A [Klick auf Tag / Label]	Erstellt einen neuen Flugplan für das angeklickte Callsign
<code>.inf</code> [Klick auf Tag / Label] oder F1 + I [Klick auf Tag / Label]	Öffnet eine neue Chatnachricht für das angeklickte Callsign; zeigt den verwendeten Pilotenclient und die aktive Frequenz des Piloten an.
<code>.contactme</code> [Klick auf Tag / Label] oder Pos1-Taste [Klick auf Tag / Label]	Schickt an das angeklickte Tag / Label die Nachricht „Contact me on ...“ mit der eigenen Primärfrequenz.

## 3. Wichtige Alias-Befehle

Befehl	Erklärung
<code>.clrs</code> (mit Startup) <code>.clns</code> (ohne Startup)	(startup approved,) cleared to \$arr via \$sid departure, flight planned route, climb <b>via SID</b> \$temp, squawk \$squawk (Platzhalter werden mit den Daten aus der DepList ersetzt)
<code>.clr</code> (mit Startup) <code>.cln</code> (ohne Startup)	(startup approved,) cleared to \$arr via \$sid departure, flight planned route, climb \$temp, squawk \$squawk

<div>.clrv (mit Startup)</div> <div>.clnv (ohne Startup)</div>	(startup approved,) cleared to \$arr via <b>vector departure</b> runway \$deprwy, after departure \$1, climb \$temp, squawk \$squawk
<div>.rbc</div>	readback correct
<div>.pbco</div>	for pushback contact \$radioname( <b>\$1</b> ) on frequency \$freq(\$1), good bye! <i>(siehe Seite oben, z. B. Stationskürzel „mp2“ einsetzen und mit TABULATOR-Taste bestätigen)</i>
<div>.co</div>	Contact \$radioname( <b>\$1</b> ) on frequency \$freq(\$1), good bye!

Befehl	Erklärung
<div>.pa</div>	Pushback approved.
<div>.pn</div>	Pushback approved, face north. <i>(Dementsprechend auch <div>.ps</div> , <div>.pe</div> , <div>.pw</div> , <div>.pne</div> , <div>.psw</div> , ...)</i>
<div>.t</div>	taxi to holding point runway \$deprwy via \$suc(\$1)
<div>.tg</div>	taxi to gate \$uc( <b>\$1</b> ) via \$uc( <b>\$2</b> )
<div>.ts</div>	taxi to stand \$uc( <b>\$1</b> ) via \$uc( <b>\$2</b> )
<div>.te</div>	taxi to entry \$uc( <b>\$1</b> ) via \$uc( <b>\$2</b> )

Befehl	Erklärung
<div>.lu</div>	line up runway \$deprwy and wait
<div>.lul</div>	behind landing traffic lineup runway \$deprwy and wait behind
<div>.lud</div>	behind departing traffic lineup runway \$deprwy and wait behind
<div>.cft</div>	wind \$winds, runway \$deprwy, cleared for takeoff
<div>.cfit</div>	wind \$winds, runway \$deprwy, cleared for immediate takeoff
<div>.ctl</div>	wind \$winds, runway \$arrwy, cleared to land

<code>.ctg</code>	wind \$winds, runway \$arrwy, cleared touch and go
<code>.cla</code>	wind \$winds, runway \$arrwy, cleared low approach
<code>.map</code>	fly missed approach procedure as published

Befehl	Erklärung
<code>.gm</code>	Good morning.
<code>.ga</code>	Good afternoon.
<code>.ge</code>	Good evening.
<code>.noatc</code>	no further ATC available, monitor UNICOM on 122.800, good bye!
<code>.bye</code>	frequency change approved, good bye!
<code>.logoff</code>	If you are not able to comply with ATC instructions (Code of Conduct B8), we encourage you to disconnect temporarily from the network. Please do not spoil the experience of other pilots and controllers.
<code>.wallop</code> [Meldung als Freitext]	Mit „.wallop“ öffnet sich ein neuer Chat in dem ein SUPgerufen und an ihn der entsprechende Text versendet wird.

Revision #4

Created 10 January 2023 14:53:22 by 1526807

Updated 10 April 2024 15:45:09 by 1586741