

TAPIMaster Connect TAPI driver

Foreword	0
Part I Content	2
1 Prepare for installation.....	3
2 Installation.....	4
3 Test Application.....	5
4 Sample projects.....	6
5 Interfaces.....	8
6 Command syntax.....	9
General information	10
Initial handshakes	11
Events	12
EventAlert.....	13
EventConnected.....	14
EventDrop.....	15
EventOffhook.....	16
EventForward.....	17
EventHold	18
EventMakeCall.....	19
Commands and responses	20
LineAnswer.....	21
LineBlindTransfer.....	22
LineCompleteTransfer.....	23
LineDial	24
LineDrop	25
LineForward.....	26
LineHold	27
LineMakeCall.....	28
LinePickup.....	29
LineRedirect.....	30
LineSetupTransfer.....	31
LineSwapHold.....	32
LineUnhold.....	33

1 Content

TAPIMaster® Connect TAPI driver 1.0 Developer Guide

- [Prepare for installation](#)
- [Installation](#)
- [Test Application](#)
- [Sample projects](#)
- [Interfaces](#)
- [Command syntax](#)

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.1 Prepare for installation

Requirements

- A computer with Windows 7 or higher and the programming language installed is required.
- Please update the operating system to the latest version.
- Download the TMConnect setup in the correct version (32 or 64 bit).
- Admin rights on the PC are required.
- TAPIMaster LineWatcher or other TAPI client program.

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

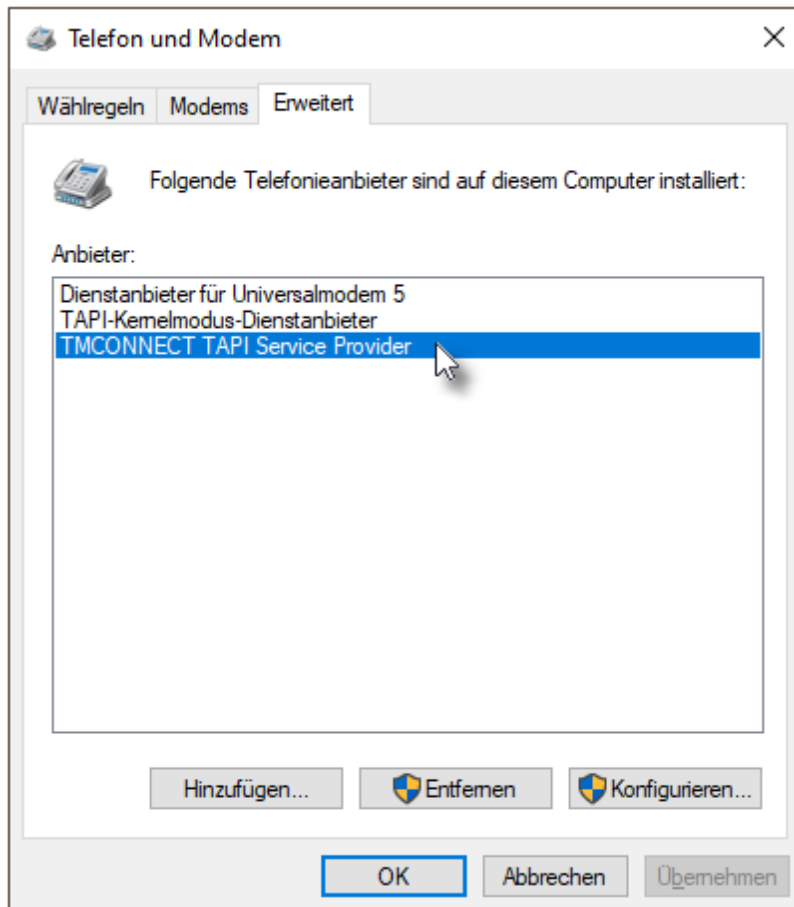
1.2 Installation

Setup

Please use the correct setup depending on the operating system (32 or 64 bit).

Start the setup and follow the instructions.

After the installation, check whether the TAPI driver is available in the phone and modem options.



[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.3 Test Application

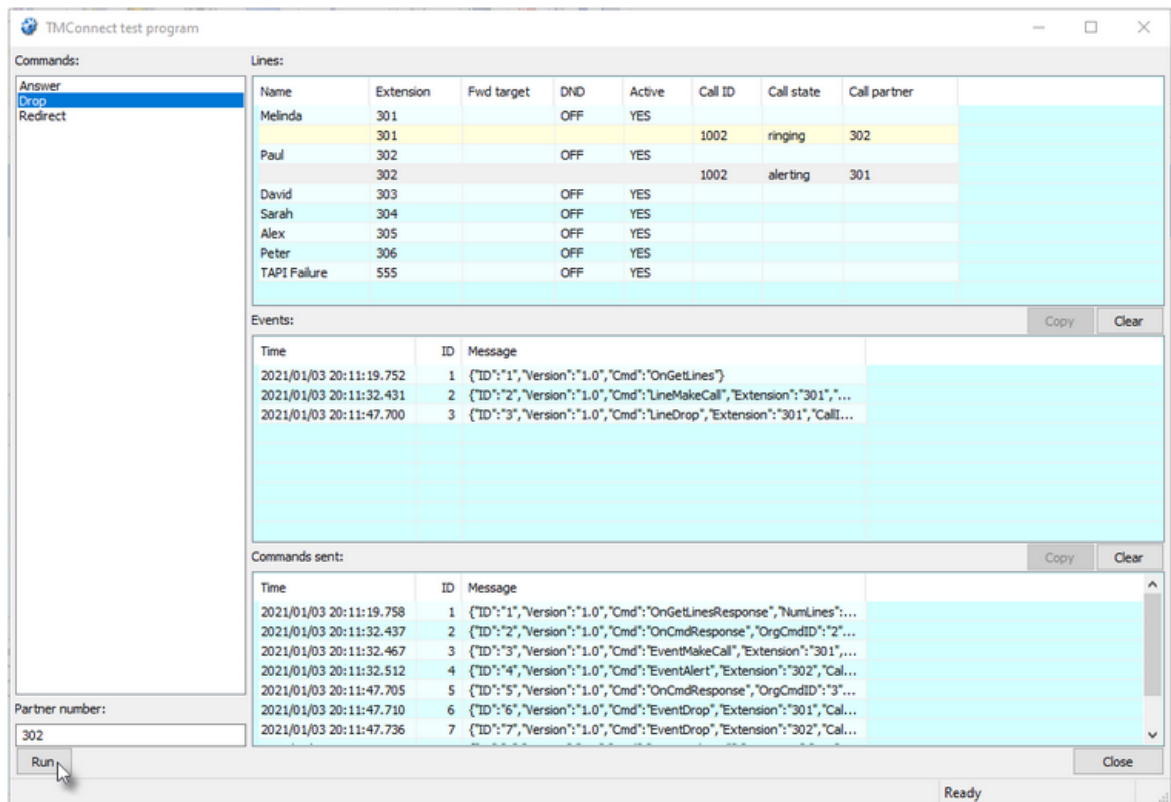
Using the test program

The setup installs a test program. This software controls the TAPI driver.

The program simulates the complete functionality which you can build into your own application.

The best practice for familiarizing yourself with the tool is to start the program with the LineWatcher and then run through scenarios with the LineWatcher.

Important: Please start the test application before using the LineWatcher.



[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

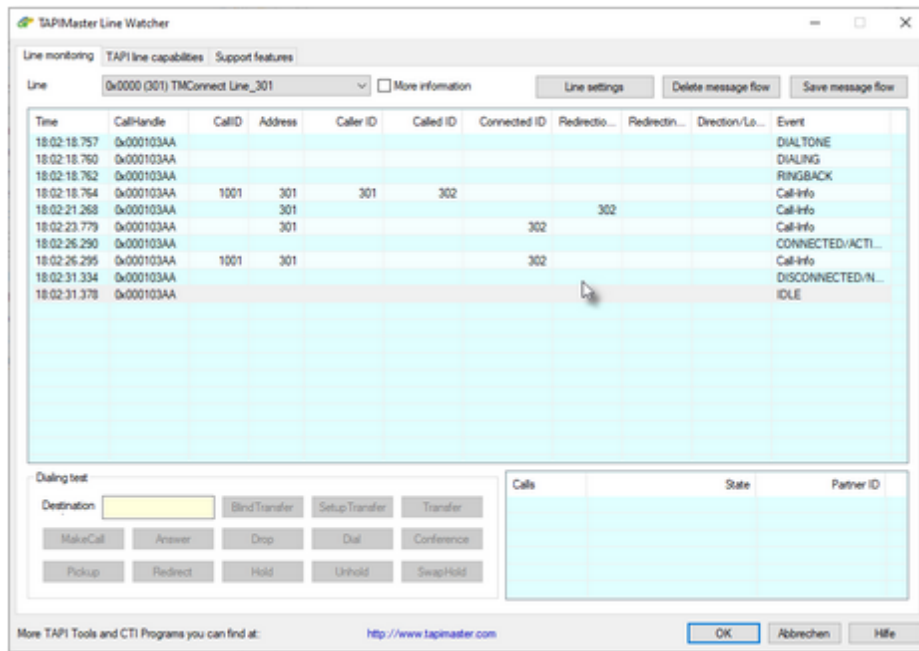
1.4 Sample projects

Using the sample application

The setup installs a small sample C# program (project TestTMConnectDotNet). Please run the TestAppTMConnect.exe as admin:



If the program is running, start the LineWatcher. Press the SendText button in the sample application. The LineWatcher will show the events.



You can find the sample in the C:\Program Files\TMCONNECT or C:\Program Files(x86)\TMCONNECT folder.

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.5 Interfaces

General infos

The TMConnect TAPI driver is addressed via an interface DLL. The software that controls the driver must be started as a Windows system service. The TMConnect test program (started with admin rights) can take over this function for test purposes..

Installed components

The TMConnect setup (or your own setup in the future) installs two files in the "C:\Windows\System32" folder: TMCONNECT.TSP and TMCONNECT.DLL. These files must have the same bit width as the operating system.

Program folder

The setup creates a program folder: C:\Program Files\TMCONNECT. In this folder, you can find different needed files for the connection to the TAPI driver:

TAPICconnector.h and *TMCONNECT.LIB*: Header file and library for embedding in C++ witten software.

TMConnect.DriverAccess.DLL: Wrapper interface DLL for .NET applications.

TMCONNECT.EXE: The TMConnect test program.

Time delays

In not licensed programs, the communication between the TAPI driver and the controlling program will be delayed. This does not affect the test program.

C++ / WIN32 API interface

This interface is very simple. This uses a callback function for receiving data and another function for sending. The DLL was built in Unicode compiler mode.

.NET interface

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6 Command syntax

[General information](#)

[Initial handshakes](#)

[Events](#)

[Commands and responses](#)

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.1 General information

The communication between the TAPI driver and the program takes place via JSON texts. These texts are the same for all interface variants.

JSON syntax

All JSON texts begin with the same fields: the message ID, the interface Version (1.0) and the Command name.

```
{
  "ID": "3",
  "Version": "1.0",
  "Cmd": "LineDrop",
  ...
}
```

Commands and responses

The TAPI driver sends commands. Every command is answered by the controlling application. The response includes the original command ID and the success state. Sample:

Command

```
{"ID": "2", "Version": "1.0", "Cmd": "LineMakeCall", "Extension": "301", "CalledID": "302"}
```

Response

```
{"ID": "3", "Version": "1.0", "Cmd": "OnCmdResponse", "OrgCmdID": "2", "Success": "1"}
```

TAPI Events

Events inform the TAPI driver about the status of devices and calls. Before implementing the TAPI commands, make sure the events are sent.

```
{"ID": "5", "Version": "1.0", "Cmd": "EventAlert", "Extension": "302", "CallID": "1001", "CallerID": "301", "CalledID": "302"}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.2 Initial handshakes

The first sequence is a request / response pair which transmits the line configuration to the TAPI driver. This request should be answered quickly.

Request

The request has no additional parameters. Cmd is OnGetLines.

```
{
  "ID": "1",
  "Version": "1.0",
  "Cmd": "OnGetLines"
}
```

Response

The response delivers the number of devices, followed from the device name an numeric extension.

```
{
  "ID": "1",
  "Version": "1.0",
  "Cmd": "OnGetLinesResponse",
  "NumLines": "7",
  "Line0": {"Name": "Melinda", "Extension": "301"},
  "Line1": {"Name": "Paul", "Extension": "302"},
  "Line2": {"Name": "David", "Extension": "303"},
  "Line3": {"Name": "Sarah", "Extension": "304"},
  "Line4": {"Name": "Alex", "Extension": "305"},
  "Line5": {"Name": "Peter", "Extension": "306"},
  "Line6": {"Name": "TAPI Failure", "Extension": "555"}
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.3 Events

Type of events

[EventAlert](#)

[EventConnected](#)

[EventDrop](#)

[EventForward](#)

[EventHold](#)

[EventMakeCall](#)

[EventOffhook](#)

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.3.1 EventAlert

Info

Signals an incoming call.

Event

Cmd: EventAlert

Extension: Internal number of the called party.

CallID: System-wide uniform identifier (integer) for the call.

CallerID: Phone number of the caller.

CalledID: The originally called number.

Sample

```
{
  "ID": "9",
  "Version": "1.0",
  "Cmd": "EventAlert",
  "Extension": "302",
  "CallID": "1002",
  "CallerID": "301",
  "CalledID": "302"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.3.2 EventConnected

Info

A call is connected.

Event

Cmd: EventConnected

Extension: Internal number of the actor.

CallID: System-wide uniform identifier (integer) for the call.

ConnectedID: Number of the call partner.

Sample

```
{  
  "ID": "11",  
  "Version": "1.0",  
  "Cmd": "EventConnected",  
  "Extension": "302",  
  "CallID": "1002",  
  "ConnectedID": "301"  
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.3.3 EventDrop

Info

A call ended.

Event

Cmd: EventDrop

Extension: Internal number of the actor.

CallID: System-wide uniform identifier (integer) for the call.

Sample

```
{  
  "ID": "14",  
  "Version": "1.0",  
  "Cmd": "EventDrop",  
  "Extension": "301",  
  "CallID": "1002"  
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.3.4 EventOffhook

Info

The handset was picket up from phone.

Event

Cmd: EventOffhook

Extension: Internal number of the actor.

CallID: System-wide uniform identifier (integer) for the call.

Sample

```
{  
  "ID": "15",  
  "Version": "1.0",  
  "Cmd": "EventOffhook",  
  "Extension": "301",  
  "CallID": "1003"  
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.3.5 EventForward

Info

A call forwarding was activated.

Event

Cmd: EventForward

Extension: Internal number of the actor. Empty if the forward state ends.

DND: on / off, DoNotDisturb state.

Target: Redirection party.

Sample

```
{  
  "ID": "17",  
  "Version": "1.0",  
  "Cmd": "EventForward",  
  "Extension": "301",  
  "DND": "off",  
  "Target": "302"  
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.3.6 EventHold

Info

A call is on hold.

Event

Cmd: EventHold

Extension: Internal number of the actor.

CallID: System-wide uniform identifier (integer) for the call.

Transfer: 1 if the PBX has created a second call in state Dialtone.

Sample

```
{  
  "ID": "26",  
  "Version": "1.0",  
  "Cmd": "EventHold",  
  "Extension": "302",  
  "CallID": "1004",  
  "Transfer": "0"  
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.3.7 EventMakeCall

Info

An outgoing call was started.

Event

Cmd: EventMakeCall

Extension: Internal number of the actor.

CallID: System-wide uniform identifier (integer) for the call.

CalledID: NUmber of the called partner.

Sample

```
{  
  "ID": "8",  
  "Version": "1.0",  
  "Cmd": "EventMakeCall",  
  "Extension": "301",  
  "CallID": "1002",  
  "CalledID": "302"  
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4 Commands and responses

Type of commands

[LineAnswer](#)

[LineBlindTransfer](#)

[LineCompleteTransfer](#)

[LineDial](#)

[LineDrop](#)

[LineForward](#)

[LineHold](#)

[LineMakeCall](#)

[LinePickup](#)

[LineRedirect](#)

[LineSetupTransfer](#)

[LineSwapHold](#)

[LineUnhold](#)

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.1 LineAnswer

Info

Answers an incoming call.

Request

Cmd: LineAnswer

Extension: Caller number

CallID: System-wide uniform identifier (integer) for the call.

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "5",
  "Version": "1.0",
  "Cmd": "LineAnswer",
  "Extension": "302",
  "CallID": "1002"
}
```

Response

```
{
  "ID": "13",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "5",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.2 LineBlindTransfer

Info

Transfers an existing call to an other party (one step transfer).

Request

Cmd: LineBlindTransfer

Extension: Internal number of the actor.

CallID: System-wide uniform identifier (integer) for the call.

Dest: Target of the call transfer.

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "17",
  "Version": "1.0",
  "Cmd": "LineBlindTransfer",
  "Extension": "303",
  "CallID": "1002",
  "Dest": "301"
}
```

Response

```
{
  "ID": "48",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "17",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.3 LineCompleteTransfer

Info

Completes a call transfer.

Request

Cmd: LineCompleteTransfer

Extension: Internal number of the actor.

HeldCallID: Unique ID of the held call (integer).

ConsultationCallID: Unique ID of the consultation call (integer).

Mode: Only mode "Transfer" is supported.

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "16",
  "Version": "1.0",
  "Cmd": "LineCompleteTransfer",
  "Extension": "301",
  "HeldCallID": "1004",
  "ConsultCallID": "1002",
  "Mode": "Transfer"
}
```

Response

```
{
  "ID": "43",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "16",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.4 LineDial

Info

Dials a number with an existing call, e. g. after picking the handset.

Request

Cmd: LineDial

Extension: Internal number of the caller.

CallID: System-wide uniform identifier (integer) for the call.

Dest: Target for the call.

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "9",
  "Version": "1.0",
  "Cmd": "LineDial",
  "Extension": "301",
  "CallID": "1003",
  "Dest": "303"
}
```

Response

```
{
  "ID": "23",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "9",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.5 LineDrop

Info

Ends a call.

Request

Cmd: LineDrop

Extension: Internal number of the actor who ended the call.

CallID: System-wide uniform identifier (integer) for the call.

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "3",
  "Version": "1.0",
  "Cmd": "LineDrop",
  "Extension": "302",
  "CallID": "1005"
}
```

Response

```
{
  "ID": "70",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "3",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.6 LineForward

Info

Enables or disables call forwarding / do not disturb mode.

Request

Cmd: LineForward

Extension: Internal number of the actor.

DND: on / off Enables / disables the do not disturb mode.

Target: Target of the call forwarding (Empty string disables forwarding).

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "22",
  "Version": "1.0",
  "Cmd": "LineForward",
  "Extension": "301",
  "DND": "off",
  "Target": "303"
}
```

Response

```
{
  "ID": "64",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "22",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.7 LineHold

Info

Put a call on hold.

Request

Cmd: LineHold

Extension: Internal number of the phone that is holding the call.

CallID: System-wide uniform identifier (integer) for the call.

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "6",
  "Version": "1.0",
  "Cmd": "LineHold",
  "Extension": "301",
  "CallID": "1002"
}
```

Response

```
{
  "ID": "16",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "6",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.8 LineMakeCall

Info

Initiates a new call.

Request

Cmd: LineMakeCall

Extension: Caller number

CalledID: Call target

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "4",
  "Version": "1.0",
  "Cmd": "LineMakeCall",
  "Extension": "301",
  "CalledID": "302"
}
```

Response

```
{
  "ID": "10",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "4",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.9 LinePickup

Info

Pick up a ringing call from another phone.

Request

Cmd: LinePickup

Extension: Internal number of the actor who picks up the call.

PickupDevice: Internal number of the telephone from which the call is picked up.

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "20",
  "Version": "1.0",
  "Cmd": "LinePickup",
  "Extension": "303",
  "PickupDevice": "301"
}
```

Response

```
{
  "ID": "57",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "20",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.10 LineRedirect

Info

Desfects an incoming unanswered call.

Request

Cmd: LineRedirect

Extension: Internal number of the actor.

CallID: System-wide uniform identifier (integer) for the call.

Dest: Deflection target.

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "19",
  "Version": "1.0",
  "Cmd": "LineRedirect",
  "Extension": "303",
  "CallID": "1002",
  "Dest": "301"
}
```

Response

```
{
  "ID": "54",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "19",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.11 LineSetupTransfer

Info

Starts a consultation call.

Request

Cmd: LineSetupTransfer

Extension: Internal number of the actor.

CallID: System-wide uniform identifier (integer) for the call.

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "8",
  "Version": "1.0",
  "Cmd": "LineSetupTransfer",
  "Extension": "301",
  "CallID": "1002"
}
```

Response

```
{
  "ID": "20",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "8",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.12 LineSwapHold

Info

Switches between two calls.

Request

Cmd: LineSwapHold

Extension: Internal number of the actor.

ActiveCallID: Unique ID of the active call (integer).

HeldCallID: Unique ID of the held call (integer).

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "15",
  "Version": "1.0",
  "Cmd": "LineSwapHold",
  "Extension": "301",
  "ActiveCallID": "1004",
  "HeldCallID": "1002"
}
```

Response

```
{
  "ID": "40",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "15",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

1.6.4.13 LineUnhold

Info

Retrieves a held state.

Request

Cmd: LineUnhold

Extension: Internal number of the telephone that retrieves the held call.

CallID: System-wide uniform identifier (integer) for the call.

Response

Cmd: OnCmdResponse

OrgCmdID: ID from the Request

Success: 1 if successful, otherwise 0

Sample

Request

```
{
  "ID": "7",
  "Version": "1.0",
  "Cmd": "LineUnhold",
  "Extension": "301",
  "CallID": "1002"
}
```

Response

```
{
  "ID": "18",
  "Version": "1.0",
  "Cmd": "OnCmdResponse",
  "OrgCmdID": "7",
  "Success": "1"
}
```

[Send feedback to TAPIMaster](#)

© 2020 Tino Kasubke. All Rights Reserved.

