AltiGen Web API Guide

October 2017

Contents

This document is provided for AltiGen Certified Partners and Admins who will be working with AltiGen's Web API.

Overview

AltiGen's Web API translates AltiLinkPlus commands via a WebProxy using RESTful-like commands. Customers can use this API to easily integrate JavaScript RESTful-like calls on their websites to provide screen pops, click-to-dial features, mute and unmute recordings, and to start, pause, resume, and end call recordings.

This document assumes that you have working knowledge of IIS and JavaScript.

Licenses

One SDK license is required for each client user.

Sessions will be released when the event channel (WebSocket) is disconnected, in order to avoid session license issues if the browser should shut down unexpectedly.

Security

When HTTP is used, a non-secure socket will be used in event channels. When HTTPS is used, a secure socket will be used instead.

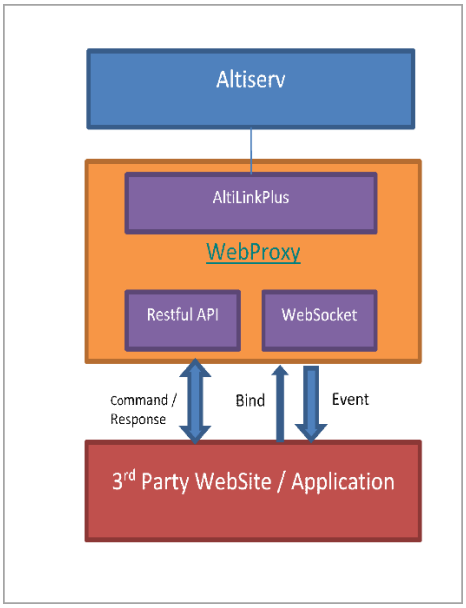
System Requirements

The AltiGen MAXC/MaxACD Web API needs only server side deployment. Make sure that your server meets the following requirements.

- Windows Server 2012 R2 or higher
- IIS 8.5
- .Net Framework 4.5 or higher

Architecture

The WebProxy resides between MAXCS / MaxACD and third-party websites or applications. It translates AltiLinkPlus commands from Altiserv to RESTful API.



There are two connections between the WebProxy and the third-party application:

- HTTP RESTful API for command / response
- WebSocket for the event

To associate the HTTP and Websocket connections for a client session, after a user logs in successfully, it should send the session ID to WebSocket to bind the event channel.

Deployment

Redundancy

The MaxACD version supports redundancy the same as with the client applications. Customers can set the server address for WebProxy as FQDN, and the WebProxy will resolve the FQDN and connect to the active server automatically. As a result, only one WebProxy is need for a redundant MaxACD system.

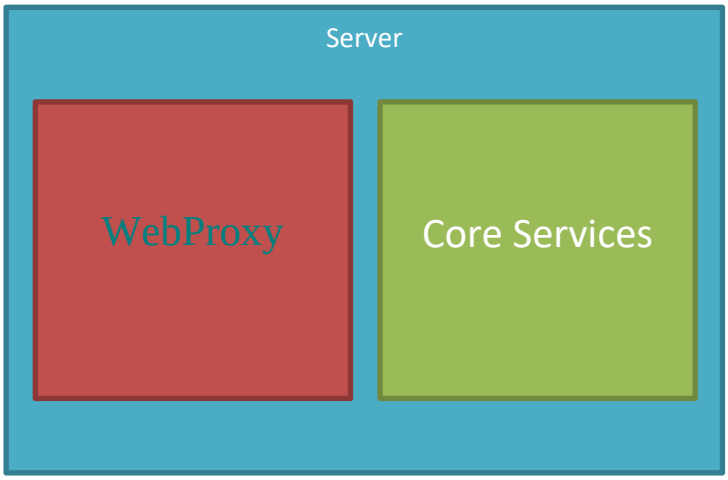
Customers can also install two WebProxy servers for a redundant MaxACD system; one for each server. Then use the FQDN to connect to each WebProxy.

Configuration Options

WebProxy can be installed either on the core server or standalone. Here are several options to deploy WebProxy.

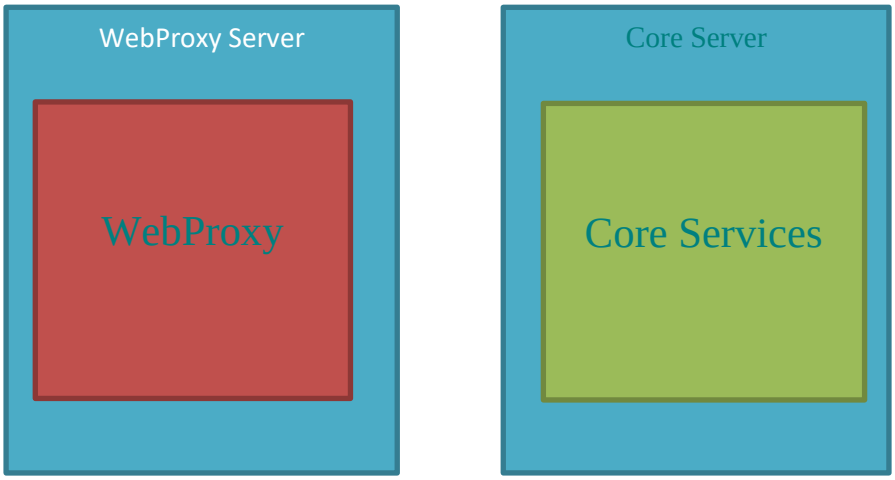
Single Sever

The WebProxy can be installed on the same machine as the core server.



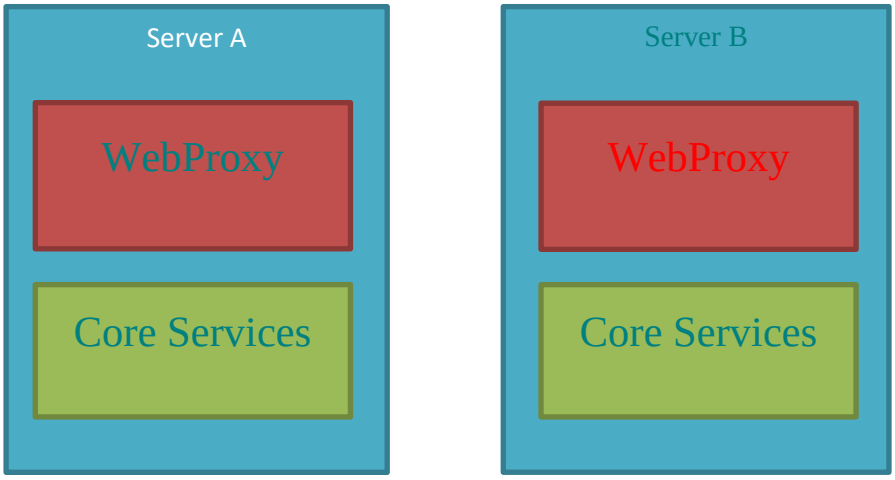
Standalone WebProxy

The WebProxy can be installed on a separate server other than the core server.



Redundant Servers

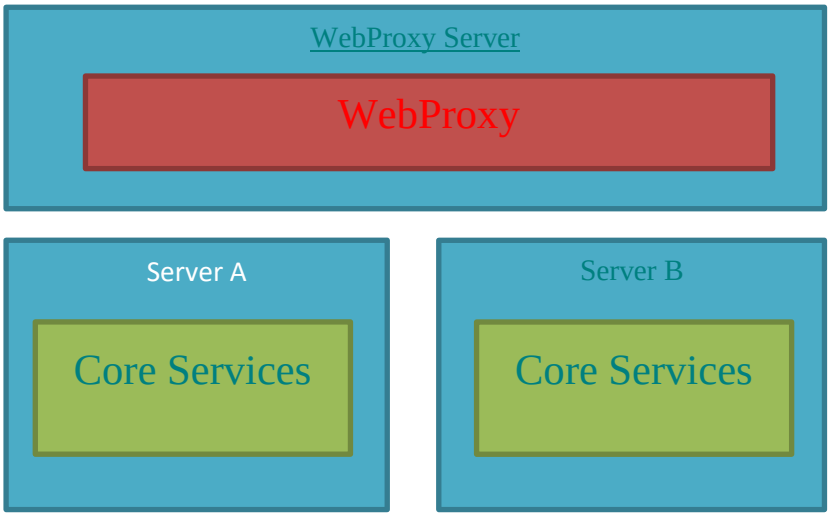
Each redundancy server can have one WebProxy installed.



Note that MaxCS does not support redundancy.

Redundant Standalone WebProxy

One WebProxy can use DNS to connect to a redundant server.



Note that MaxCS does not support redundancy.

Installation Notes

WebProxy is a standalone installation. It can be installed on a core MAXCS or MaxACD server, or on a separate server.

When you install WebProxy, make sure that you install it to a site that binds to a port other than port 80, because port 80 is already reserved for the AltiGen Polycom Web Configuration service.

Web API Commands

Every web API needs an http method and a URL. For example, Logon: POST /sessions

Logon

POST /sessions

Summary

Log in the user. Basic authentication will be used. WebProxy supports single user multiple location login.

Parameters

Name: session

Located in: body

Description: existing session ID

Required: no

Schema:

```
Session {
  sessionId:   string
}
```

Responses

Ok: returns existing session id if found on server, or else creates a new session

Unauthorized: the username / password does not match

PaymentRequired: no license

Forbidden: rejected by the server

InternalServerError: an unexpected error has occurred

Log Off

DELETE /sessions/{sessionId}

Summary

Logs off the user

Parameters

Name: sessionId

Located in: path

Description: existing session ID

Required: yes

Schema:

Responses

Ok: Logoff was successful

InternalServerError: an unexpected error has occurred

```
Error {
  code:  integer
  message:  string
}
```

Make Call

POST /sessions/{sessionId}/calls

For number format policy, refer to libphonenumber: https://github.com/googlei18n/libphonenumber.

Summary

Places a call

Parameters

Name: sessionId

Located in: path

Description: existing session ID

Required: yes

Name: target

Located in: body

Description: call target

Required: no

Schema:

```
CallTarget {
  trunkAccessCode:      string
                        trunk access code. If this value does not
                        exist, route access code will be used. (Only
                        applies to MaxCS.)
```

```
  target:               string
                        target number. In MAXCS it is PSTN number.
                        In MaxACD, it is PSTN number or SIP URI.
}
```

Responses

Ok: the call was successful

NotFound: the session could not be found

InternalServerError: an unexpected error has occurred

```
Error {
  code:   integer
  message:    string
}
```

Drop Call

DELETE /sessions/{sessionId}/calls/{callHandle}

MaxACD does not support the Drop Call API.

The Drop button will not work for a conference call.

Summary

Drops the current call

Parameters

Name: sessionId

- Located in: path
- Description: existing session ID
- Required: yes
- Schema: string

Name: callhandle

- Located in: path
- Description: call handle
- Required: yes
- Schema: string

Responses

Ok: the call was successfully terminated

NotFound: the session or call could not be found

InternalServerError: an unexpected error has occurred

```
Error {
```

```
  code:     integer
  message:  string
}
```

Record Call

POST /sessions/{sessionId}/record

Summary

Records the call.

When the user mutes and unmutes the call, the recording file continues. A short beep tone will record while the user is muted. The mute and unmute actions need to be at least 500ms apart. Note that mute / unmute is not supported in MaxACD; use Pause instead.

The beep tone and the mute do not change the length of the recorded file.

Parameters

Name: sessionId

Located in: path

Description: existing session ID

Required: yes

Schema: string

Name: callhandle

Located in: path

Description: call handle in call event

Required: yes

Schema: string

Name: data

Located in: body

Description: record data

Required: yes

Schema: string

```
RecordData {
  Operation: string
                Start / Mute / Unmute / Stop
  filename:     string
                record destination file on server. If blank,
                server will generate it
}
```

Responses

Ok: the recording was successful

NotFound: the session or call could not be found

InternalServerError: an unexpected error has occurred

```
Error {
  code:   integer
  message:    string
}
```

Web API Events (Web Socket)

The web socket is used to dispatch events, including call events and recording events. Existing calls will be sent as an event when the web socket is first established.

CallEvent

```
CallEvent {
 eventType:        string
                   CallEvent
 callHandle:       int32
                   call pad
 callEventType:    string
                   ring/ringback/connect/disconnect
 direction:        string
                   in or out
 callInfo:         callInfo  {
                   call info
                   callerID:                    string
                                                callerID
                   callerName:                  string
                                                caller name
                   callerIPaddress:             string
                                                caller IP address
                   calleeID:                    string
                                                callee ID
                   calleeName:                  string
                                                callee name
                   ivrData:                     string
                                                ivr data
                   userData:                    string
                                                user data
                   workgroupNumber:             string
                                                workgroup number
                   outboundworkgroupNumber:          string
                                                outbound workgroup number
                   dnisName:                    string
```

```
                                    dnis name
                dnisNumber:         string
                                    dnis number
                aniName:            string
                                    ani name
                aniNumber:          string
                                    ani number
                callPriority:       integer
                                    call priority
                callSklr:           integer
                                    call sklr
                callType:           string
                                    callback call or not
                originalData:       originalData {
                                    original call data for callback workgroup call
                                    originalworkgroupNumber     string
                                                                original workgroup number
                                    originalCallerId            string
                                                                original caller ID
                                    originalCallername          string
                                                                original caller name
                                    }
                }
}
```

RecordEvent

RecordEvent will be sent even in auto-record state.

```
RecordEvent  {
    eventType:          string
                        RecordEvent
    CallHandle:         int32
                        call pad
    recordingState:  string
                        recording state, Start / Stop / Mute
    recordType:         string
                        Agent / Supervisor / System
}
```

ServerStatusEvent

```
ServerStatusEvent  {
    eventType:          string
                        ServerStatusEvent
    Connected:          boolean
```

```
                        is webproxy connected to AltiServ
}
```