Transfer

The `Transfer` voice element performs a call transfer to a phone number specified by a configuration setting. Depending on how the voice browser is configured, the call transfer can be a bridge transfer or a blind transfer. For a bridge transfer, the voice browser makes an outbound call while maintaining the original call and acts as a bridge between the two calls. The advantage of this is that once the secondary call ends, the original call can still continue with the IVR. The disadvantage is that two separate phone lines are used. For a blind transfer, the voice browser makes an outbound call and when connected, links the original call to the new caller through the use of a telephony switch. At this point, the voice browser (and as a result VXML Server) is no longer in control of the call. Blind transfers involve only one line.

The Transfer element defines exit states for the different ways bridge transfers can end such as the person being called hung up, there was no answer, there was a busy signal, or some other phone-related error occurred. Since blind transfers take the call away from the voice browser and VXML Server, a Transfer element performing a blind transfer would never return an exit state. Instead, an special event would be thrown by the voice browser, caught in the root document for the call, and VXML Server would terminate the session by interrupting the Transfer element.

The number to transfer to can be any phone number allowed by the voice browser telephony provider (some may place restrictions on outbound dialing). Please note that different voice browsers may or may not accept certain kinds of phone numbers. Check your voice browser documentation for specific requirements and restrictions for call transfer.

Settings

Name (Label)	Type	Req'd	Single Setting Value	Sub. Allowed	Default	Notes

transfer_destination (Transfer Destination)	string	Yes	`true`	`true`	*None*	The phone number to transfer to. It may contain non-numerical characters to allow support for phone extensions. If the destination_type is `sip`, make sure that the value for transfer_destination is in the SIP URI (number@domain) format.
destination_type (Destination Type)	string	No	`true`	`true`	tel	The type of transfer destination to which the voice element is to connect. Possible values are: `tel` | `sip`.
connect_timeout (Connect Timeout)	string	Yes	`true`	`true`	60s	The maximum time (in seconds) that voice element is allowed to wait for an answer, before exiting with a *noanswer* exit state. Possible values are standard time designations including both a positive integer and a time unit s, for example, 10s (for 10 seconds). Default = 60s.
max_transfer_time (Max Transfer Time)	string	Yes	`true`	`true`	0s	The maximum duration (in seconds) that the transfer is allowed to last. Possible values are standard time designations including both a non-negative integer and a time unit s, for example, *30s* (for 30 seconds). Default = 0s (means no limit). This setting only applies when `bridge` is set to *true*.
bridge (Bridge)	binary	Yes	`true`	`true`	false	Determines whether the application remains connected to the caller after the transfer is initiated. Possible values are: `true` | `false`. Default = *false*. When set to *false* (that is, a blind transfer), the application redirects the caller to the callee without remaining in the connection; the transfer outcome is completely unsupervised. When set to *true* (that is, a bridge transfer), the application stays connected to the caller and adds the callee to the connection for the duration of the transferred call.
transfer_audio (Transfer Audio)	string	No	`true`	`true`	*None*	The URI location of the audio file to be played while connecting the call.
aai (Application-to-application Information)	string	No	`true`	`true`	*None*	A string containing Application-to-Application Information data to be sent to an application on the far-end.

Element Data

Name	Type	Notes
result	string	The value returned by the transfer field. This is dependent on the voice browser.

Exit States

Name	Notes
busy	The number was busy.
noanswer	There was no answer.
phone_error	There was some sort of phone-related error.
done	The call transfer completed successfully.

Note	Hosting voice browsers may disable call transfers for developer accounts. You should verify with your provider that transfer is enabled for your application.
Note	Some voice browsers use a code to indicate which call transfers will be allowed. This code appears before the phone number.
Note	Some voice browsers support the inclusion of an extension in the phone number so that the system can transfer to a particular extension. It is up to the developer to pass this voice element a string containing the appropriate format. Check the platform specific documentation for support of extension dialing in transfer.

Audio Groups

Transfer Audio

Name (Label)	Req'd	Max1	Notes
initial_audio_group (Initial)	No	Yes	Played to introduce the transfer. If there is none, the transfer occurs immediately.
busy_audio_group (Busy)	No	Yes	Played when there is a busy signal, right before the voice element exits with the "busy" exit state.

noanswer_audio_group (No Answer)	No	Yes	Played when there is no answer, right before the voice element exits with the *noanswer* exit state.
phone_error_audio_group (Phone Error)	No	Yes	Played when there is some kind of phone-related error, right before the voice element exits with the *phone_error* exit state.

End

Name (Label)	Req'd	Max 1	Notes
done_audio_group (Done)	No	Yes	Played when the call transfer completes with the party called hanging up and the caller staying on the line.

Folder and Class Information

Studio Element Folder Name	Class Name
Call Control	com.audium.server.voiceElement.transfer.MTransfer

Events

Name (Label)	Notes
`Event Handler`	You can select either VXML Event or Java Exception as event handler type from the drop-down list.