# Voice Callback and URS

## VCB Notification Structure

12/12/2025

## Contents

# VCB Notification Structure

A VCB notifiable call must have in its extensions the `notifyurl` key. This key, together with a few other optional extensions, controls the location and format for sending VCB notifications.

The additional keys are:

- `notifybody`
- `notifyenc`.

The logic for sending a VCB notification is as follows:

## For an ORS Instance

- If `notifyurl` starts with `ors://` then the entire `notifyurl` should be in the format, `ors://orsname/scxml/session/orssession/event/eventname[?params]`, and URS will try to send an event to the specified ORS and the specified session within it.
- If `notifyurl` starts with `ors:` then the entire `notifyurl` should be in the format, `ors:whatever/event/eventname[?params]`, and URS will try to send an event to the ORS node and the session associated with this VCB call.

In both the above cases, the optional `params` and `notifybody` can be provided.

- `params` should have the format of an URL encoded string of parameters.
- `notifybody` could either be an URL encoded string of parameters or a JSON string.

Their values if provided will be decoded first – any fragment in square brackets will be replaced with its actual value as given below:

| | |
|---|---|
| [udata] | entire call attached data (& separated) |
| [udata.*] | entire call attached data (, separated) |
| [udata.key] | value of corresponding attached data |
| [udataj] | entire call attached data as JSON string |
| [ext] | all call extensions (& separated) |
| [ext.*] | all call extensions (, separated) |
| [ext.key] | value of corresponding extension key |
| [extj] | all call extensions as JSON string |
| [orssession] | ORS session ID if call has associated ORS session ID |
| [ors] | `host:port` of ORS node associated with this VCB call |
| [call.connid] | Connection ID of this VCB call |

| | |
|---|---|
| [call.uuid] | UUID of this VCB call |

## For an URS Instance

- If `notifyurl` starts with `urs://` then the entire `notifyurl` should be in the format, `urs://ursname/message` and URS will try to send a command to the specified URS.
- If `notifyurl` starts with `urs:` then the entire `notifyurl` should be in the format, `urs:message` and the particular URS will try to send a command to itself.

In both cases, the command is effectively invoking the `RequestRouter` function (updated description in the Supplement to the Universal Routing 8.1 Reference Manual):

```
RequestRouter[ursname, message, notifybody, ","]
```

Both `message` and `notifybody` preliminary will be decoded as described above.

## For a Generic HTTP Server

In all other cases, URS will try to send a VCB notification as a REST HTTP message.

- `notifyurl` must be a valid URL.
- `notifybody` – If specified, URS will use POST message instead of GET message, and use `notifybody` as the content of this POST message.
- `notifyenc` – Used only if `notifybody` is present and will be used as the content of Content-Type header of the generated HTTP message.

Both `notifyurl` and `notifybody` preliminary will be decoded as described above.