

# vrExpress

vrExpress *agent-id addressee-id utterance-id xml-message*

## Description

Sent from an agent, Dialogue Manager or the [NPCEditor](#) to the [NonVerbal Behavior Generator](#). Usually containing the line the character will say and for which nonverbal behavior needs to be generated, or more high level instructions on what type of behavior to generate:

- speech
- posture change
- status / request
- gaze
- emotion
- listener feedback
- coping stance
- negotiation stance
- COA

## Parameters

- *agent-id*, ID of the virtual human (i.e. 'ChrBrad')
- *addressee-id*, ID of the addressee (i.e. 'user')
- *utterance-id*, unique ID within a session
- *xml-message*, the text to be turned into audio, in XML format (see below for examples). This is a mix between Functional Markup Language (FML) and Behavior Markup Language (BML). The general structure of the *xml-message* comprises of the following -
  - *xml-header* which is generally the same to specify the beginning of the xml block. It is usually `<?xml version="1.0" encoding="UTF-8" standalone="no" ?>`
  - *act-tag* which is always the same and is written as `<act>`. This denoted the the start of the fml / bml act and is required as a part of the fml / bml standard.
  - *participant-tag* specifies the participant id (usually same as the *agent-id*) and the rold of the participant. It is usually something like `<participant id="harmony" role="actor" />`
  - *fml* is the Functional Markup Language parsed by NVBG for behavior generation. The examples of FML can be seen below. This tag may be empty if there is no FML present.
  - *bml* is the Behavior Markup Language and is usually forwarded to the next module like smartbody. Generally, after the FML is parsed by NVBG, it creates the BML and adds to this tag.

## Examples

### Speech

The speech messages are characterized by the speech tag within them. They are interpreted and the corresponding output bml is generated with the speech time marks, animations, head-nods, facial-movements etc. These animations are generated based on the content of the speech tag and the fml tag in the input message.

```

vrExpress harmony ranger harmony221 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <act>
    <participant id="harmony" role="actor"/>
    <fml>
      <intention>
        <object name="A316">
          <attribute name="addressee">ranger</attribute>
          <attribute name="speech-act">
            <object name="A317">
              <attribute name="content">
                <object name="V28">
                  <attribute name="modality">
                    <object name="V29">
                      <attribute name="conditional">should</attribute>
                    </object>
                  </attribute>
                </object>
              </attribute>
            <attribute name="polarity">negative</attribute>
            <attribute name="attribute">jobAttribute</attribute>
            <attribute name="value">bartender-job</attribute>
            <attribute name="object-id">utah</attribute>
            <attribute name="type">state</attribute>
            <attribute name="time">present</attribute>
          </object>
        </attribute>
        <attribute name="motivation">
          <object name="V27">
            <attribute name="reason">become-sheriff-harmony</attribute>
            <attribute name="goal">address-problem</attribute>
          </object>
        </attribute>
        <attribute name="addressee">ranger</attribute>
        <attribute name="action">assert</attribute>
        <attribute name="actor">harmony</attribute>
      </object>
    </attribute>
  </object>
  </intention>
</fml>
<bml>
  <speech id="spl" type="application/ssml+xml">ranger utah cant be bartender if he
becomes sheriff</speech>
</bml>
</act>

```

### Posture Change

These messages are characterized by the <body posture=""> tag which allows NVBG to know that there has been a change in posture.

```

vrExpress harmony None XX <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <act>
    <participant id="harmony" role="actor" />
    <bml>
      <body posture="HandsAtSide" />
    </bml>
  </act>

```

### Status / Request

The idle\_behavior and all\_behavior attributes within the request tag allows NVBG to keep track of whether or not to generate the corresponding behavior.

```
vrExpress harmony None XX <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <act>
    <participant id="harmony" role="actor" />
    <fml>
      <status type="present" />
      <request type="idle_behavior" value="off" />
    </fml>
  </act>
```

## Gaze

These gaze tags, if present within the input message are transferred unaltered to the output message.

```
vrExpress harmony ranger constant103 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <act>
    <participant id="harmony" role="actor" />
    <fml>
      <gaze type="weak-focus" target="ranger" track="1" speed="normal" > "listen_to_speaker" <
/gaze>
    </fml>
  </act>
```

## Emotion

The affect tag contains data about the emotional state the character is currently in. This can be used to affect output behavior.

```
vrExpress harmony None schererharmony17 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <act>
    <participant id="harmony" role="actor" />
    <fml>
      <affect type="Fear" STANCE="LEAKED" intensity="110.475"></affect>
    </fml>
  </act>
```

## Listener Feedback

The listener feedback is generally used by the virtual agents for active listening while the subject is speaking. This can be used to give behavior feedback while listening.

```
vrExpress harmony None listenerFeedbackharmony14 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
  <act>
    <participant id="harmony" role="actor" />
    <fml>
      <listenerFeedback speaker="ranger" polarity="positive" agreement="neutral" uttid="gsym4"
>
      </listenerFeedback>
    </fml>
  </act>
```

## Coping Stance

The coping stance is generally used by the virtual agent to cope with a change in state.

```

vrExpress harmony None copeharmony20 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
    <act>
        <participant id="harmony" role="actor" />
        <fml>
            <affect type="Anger" STANCE="LEAKED" target="captain-kirk"></affect>
            <affect type="Anger" STANCE="INTENDED" target="negotiate-become-sheriff-harmony"><
/affect>
                <coping type="procrastinate" target="negotiate-become-sheriff-harmony"></coping>
            </fml>
        </act>

```

## Negotiation Stance

The negotiation stance tags as the name suggests are used by virtual agents to specify their negotiation stance on a specific plan.

```

vrExpress harmony None stanceharmony21 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
    <act>
        <participant id="harmony" role="actor" />
        <fml>
            <negotiationStance type="delay" coa="become-sheriff-harmony" >
            </negotiationStance>
        </fml>
    </act>

```

## COA

The COA tags are used by the virtual agents to specify their Course of Action.

```

vrExpress harmony None coaharmony22 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
    <act>
        <participant id="harmony" role="actor" />
        <fml>
            <COA coa="become-sheriff-harmony" >
            </COA>
        </fml>
    </act>

```

## Sending Components

- [NPCEditor](#)

## Receiving Components

- [NonVerbal Behavior Generator](#)

## Related messages

- [vrSpeak](#)
- [vrAgentBML](#)