

## Memo



**To:** SpeechStorm Customers  
**From:** Michael Orr (m.orr@speechstorm.com)  
**Date:** 17 July 2012  
**Ref:** MO120717.1.doc

### Identification Data Contracts

This memo describes the data contract that is used by the SpeechStorm Identification product. Each of the template callflows provided with this product have a single integration point near the end of the flow, the purpose of which is to lookup customer details based on the information that has just been provided by the caller.

This document corresponds to version 1.0 of the SpeechStorm3 Identification product.

### General SpeechStorm3 Integration Principles

Integration points in SpeechStorm take the form of simple web services which pass parameters as key-value pairs using a HTTP POST request, and which expect a basic XML message to be returned in the response body.

The URL for the web service is configured in the SpeechStorm Control Centre.

The parameters that are passed in the POST request include both product-specific values – such as the payment card number in Payment Capture, or the postcode that was spoken by the caller in Address Capture – and any other variables or attached data that are currently in session.

There are three additional parameters that are passed: “cli” is the number the user is calling from; “dnis” is the number that they dialled; and “sessionid” is the VXML platform session ID.

The exact contents of the XML response will be different for different products, but will follow some general conventions across all SpeechStorm products. Most products’ web services allow you to create or set the value of variables via the response XML. When specifying variables in the response, you can cause some or all key-value pairs to be attached to the call via CTI (where the platform supports it) by including an optional “attach” attribute with a value of “true”. Similarly, you can choose to have some or all variables added to the caller’s CLI Data record by including an optional “remember” attribute with a value of “true”.

### Integration Point #1: Perform Lookup Web Service

This is the point where the caller’s details are checked against the client’s backend systems: it passes the answer(s) to the security questions that have just been asked and receives a notification to say whether the answers are valid, along with any other arbitrary key-value pairs that may be of use to the application.

## Request

The request will be an HTTP POST request, passing the details of the n-best list for each question, as well as any variables that are currently in session. ("N-best" refers to the multiple possible matches that might be returned if speech recognition is used, each with a "confidence score" to indicate how good a match it is. If desired, this can be used for "fuzzy matching" but it is also acceptable to just use the first item in the n-best list: i.e. the best match.)

The details for each question will include an "nbestcount" parameter plus one or more "nbest/*N*" and "confidence/*N*" parameters, where *N* is the n-best index starting at 1.

The standard request parameters are as follows:

Parameter Name	Description	Example Values
questioncount	The number of questions that are asked in the identification attempt	Will be "1" for the "Single Question Template"
questionX_nbestcount	The number of nbest entries for the Xth question	"2"
questionX_nbest/ <i>N</i>	The answer, or "meaning", that was recognised in <i>N</i> th place on the n-best list for the Xth question	"Brazil", "2244499"
questionX_confidence/ <i>N</i>	The confidence score between 0 and 1000 for the <i>N</i> th place on the n-best list for the Xth question	"469", "980", "1000"

A typical request POST URL might look like:

<http://myserver:8080/myservices/test/LookupCustomerDetails.jsp>

And a typical HTTP POST body might look like:

```
questioncount=1&question1_nbestcount=2&question1_nbest1=123456&
question1_confidence1=980&question1_nbest2=332222&question1_confidence2=604&
cli=02890571100&sessionid=1234%2D3AAF%2D3372
```

## Response

The XML response specifies the overall status of the identification attempt (i.e. "success", "not found", or some other return code such as "error"), and provides a mechanism to set arbitrary variables in the SpeechStorm call session.

In any of these XML responses, more than one <variable> element can be included if desired, or the entire <variables> structure can be omitted completely.

A typical XML response where the caller's answers to the question(s) are deemed to be correct would look like this:

```
<identificationResult>
  <status>success</status>
  <variables>
    <variable name="AccountNumber " value="12345678" attach="true" />
    <variable name="CurrentBalance" value="GBP24.45" />
    <variable name="CustomerSegment" value="Gold" />
  </variables>
</identificationResult>
```

A lookup where the caller's answers did not match an existing customer account would look like this:

```
<identificationResult>
  <status>not found</status>
</identificationResult>
```

If an error occurs then the response might look like this:

```
<identificationResult>
  <status>error</status>
  <variables>
    <variable name="WhatWentWrong" value="Backend system timed out" />
  </variables>
</identificationResult>
```

#### Notes:

- Only the <status> element is mandatory. As well as signalling the success (or otherwise) of the lookup, it can be used to drive the callflow. For example:
  - If the <status> is "success" then the "success" path will be followed in the callflow.
  - If the <status> is "error" then the application's error handling path will be followed in the callflow.
  - If the <status> is "agent" then the application's 'agent' path will be followed in the callflow.
- Status should be "success" to indicate that the lookup was successful and the details provided by the caller are valid.  
A status of "not found" means that the details provided by the caller were not valid and that the caller should be given another attempt (up to the configured maximum).
- Any other status causes the callflow module to 'return' with that status.
- HTTP response codes other than "200" will be treated as an error and the module will return with a result of "error".
- Optional attributes can be added to <variable> elements:
  - Adding `attach="true"` will cause the variable to be attached to the call (CTI)
  - Adding `remember="true"` will cause the variable to be stored against the caller's CLI (if the CLI is deemed to be unique) and can thus be made available on subsequent calls from the same CLI