

# Hello World Service

---

## Introduction

The Hello World service is a simple example service that changes the caller's caller ID name to a friendly greeting when inserted into a call flow. It exists to demonstrate the structure of a very simple service, and to allow for the testing of new Avaya Breeze® platform deployments.

## Overview

The Hello World service is inserted on both the calling side and called side for calls made by users that have the service enabled in their Service Profile.

When a call enters the Hello World service, the service looks up the `displayString` attribute from the calling user's Service Profile, as configured in the Avaya Aura® System Manager, and uses that value to change the calling user's caller ID name.

## Concepts Demonstrated

- Intercepting calls
- Reading an attribute from a user's Service Profile
- Changing a caller's display name

## Detailed description

The service defines an attribute `displayString` which is configurable on a per-Service-Profile basis in the Avaya Aura® System Manager UI. The attribute is declared in the `properties.xml` file located at `helloservice-svar/src/main/resources/properties.xml`. The following snippet of the `properties.xml` file defines the `displayString` attribute:

```
<attribute name="displayString">
  <displayName>Display String</displayName> <helpInfo>String
  used for caller's caller ID display</helpInfo>
  <global>false</global>
  <validation name="anyString">
    <type>STRING</type> </validation>
  <admin_visible>true</admin_visible> <factory>
    <value>Hello from Avaya Breeze</value>
    <user_changeable>true</user_changeable>
  </factory>
</attribute>
```

For detailed information about service attributes, refer to the *Service Development Guide*.

The Java source for the Hello World sample service is contained in only one class, `HelloServiceListener` in `helloservice-war/src/main/java/com/avaya/zephyr/services/helloservice/`. This class extends the `CallListenerAbstract` class, which internally implements the `CallListener` interface. Additionally `HelloServiceListener` is annotated with `TheCallListener`. Because `HelloServiceListener` implements the `CallListener` interface (implicitly through its extension of the `CallListenerAbstract` class) and is annotated with `TheCallListener`, the Avaya Breeze framework will invoke this class's `callIntercepted` method when a new call triggers the service. The `callIntercepted` method first retrieves the calling party and the called party from the `Call` object and writes a message to the debug log. Then, it declares the variable `displayFromServiceProfile`, which will be used to read the value of the `displayString` attribute mentioned above.

What happens next depends on a characteristic of a call referred to as the phase. In Avaya Breeze a call is treated as consisting of two parts, the calling phase and the called phase. Consequently, a service is defined to be invoked for either the calling side of a call or the called side of a call or both. The question then becomes, What determines if a particular action is a calling side action or a caller side action? In general, most actions seem to more naturally be associated with the calling or called party. For example, imagine a company where there are limitations on who is allowed to make calls from company phones to external numbers. People in certain roles are allowed to make calls while others are not. A service written to enforce such limitations would probably be defined as a calling phase service, since the decision to enforce the limitation is primarily based on the calling party extension.

However, other factors can also affect how a service is defined. And the Hello World service illustrates another justification for how to designate a service.

A look in the `properties.xml` file reveals that the Hello World service is defined to be both a calling and called phase service, as shown here:

```
<orig_order>1</orig_order>
<orig_group>1</orig_group>
<term_order>1</term_order>
<term_group>1</term_group>
```

The reason for this approach was very practical. The Hello World service must act both on calls coming from external extensions to phones internal to the company as well as on calls going from internal phones to external extensions. Further, the only way to configure a service to be invoked is if the profile containing a service is assigned to internal users. Therefore, for calls leaving the company, the service must be invoked as a calling phase service; conversely, calls into the company must invoke the service as a called phase service.

But regardless of which phase the service is invoked for, the Hello World service always modifies the display name of the calling party.

Therefore, the next thing that the code does is branch based on whether the service has been invoked for the calling phase or the called phase. If we are in the calling phase, the value of the `displayString` attribute is retrieved using the `callingParty` information; if we are in the called phase, the value is retrieved using the `calledParty` information.

Next, assuming a non-null value was retrieved for the attribute, the display name of the caller is set to that value using the method `setPresentedDisplayName`. Finally, a call to the `Allow` method of the `Call` object directs the framework to continue the call set up. Note that the call to the `Allow` method is optional, since this is the default behavior if no other call action (like “drop” or “divert to” or similar) is specified before exiting the `callIntercepted` method.

## Installation and Configuration

For information on installing the service, assigning it to users, and configuring the display string attribute for Service Profiles see the guide *Installing, Configuring and Testing an Avaya Breeze Service*.

## Testing the service

Assume there are two users, A and B, where A is a caller external to your company and B is a caller internal to your company.

- Make a call from A to B. Note the caller ID information.
- Enable the Hello World service in user B’s Service Profile, and wait about a minute for data replication.
- Make another call from A to B. Notice that A’s caller ID information has been replaced with a friendly greeting on B’s phone.
- Note that you can also call from B to A to exercise the service. However, because of limitations with some telephone providers on the public network, the display name often will not pass through the public network to the called phone. Therefore, it may not be possible to see the effect of this service for calls of this type.

## Troubleshooting PROBLEM:

Caller ID display not changed on called party’s phone.

ACTION:

Verify that the service is being invoked by running `ce dlogw | grep “Hello World”` on the Avaya Breeze server and making calls from users for whom the service is enabled. If the service is being executed, you should see log messages like `Call from A to B entered Hello World`

Service in the output. Otherwise, ensure that the service is properly installed and enabled for the calling user.

For more information on troubleshooting Avaya Breeze services, refer to the *Troubleshooting Guide*.