



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Callback User's Guide

Exception Patterns

5/10/2025

Contents

- [1 Exception Patterns](#)
 - [1.1 Customer Phone Numbers](#)
 - [1.2 Create a Pattern Group](#)
 - [1.3 Create an Exception Pattern](#)
 - [1.4 Test your Exception Pattern](#)
 - [1.5 Add the Pattern Group to your Callback Service](#)
 - [1.6 How to get Help about Supported Patterns?](#)

Exception Patterns

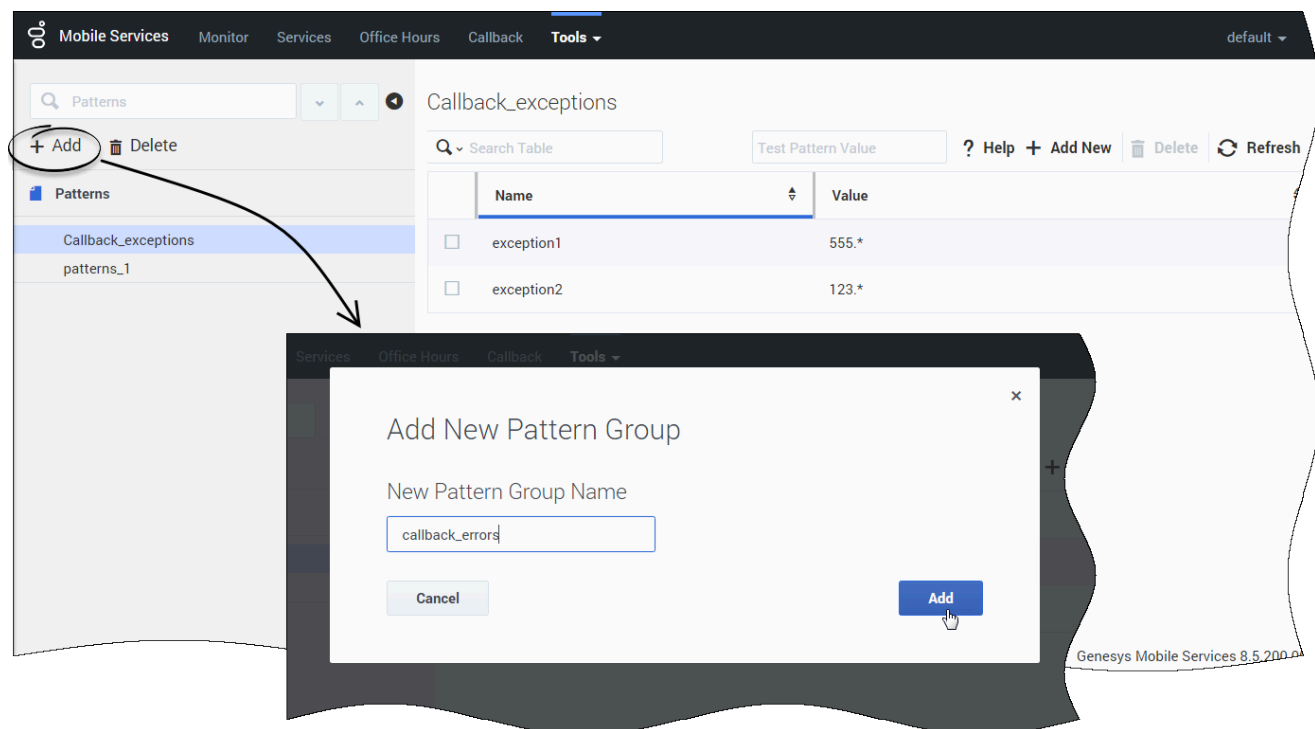
Genesys Callback allows implementing patterns to be checked before performing the callback attempt. The **Patterns** tab enables you to configure exceptions; for example, phone numbers. You can define as many patterns as you wish, then add them to your Callback services.

Customer Phone Numbers

For the `_customer_number` parameter used in Callback, note that the system internally adds a + sign to the phone number by default. As a result, if `_customer_number=12345678901`, it will become `_customer_number=+12345678901` in the system before going through the processing of the patterns exception. So, to define patterns exception for specific phone numbers, you need to add the + sign at the start of your pattern exception, for example:

```
[cb_errors]
12345678901=\+12345678901
```

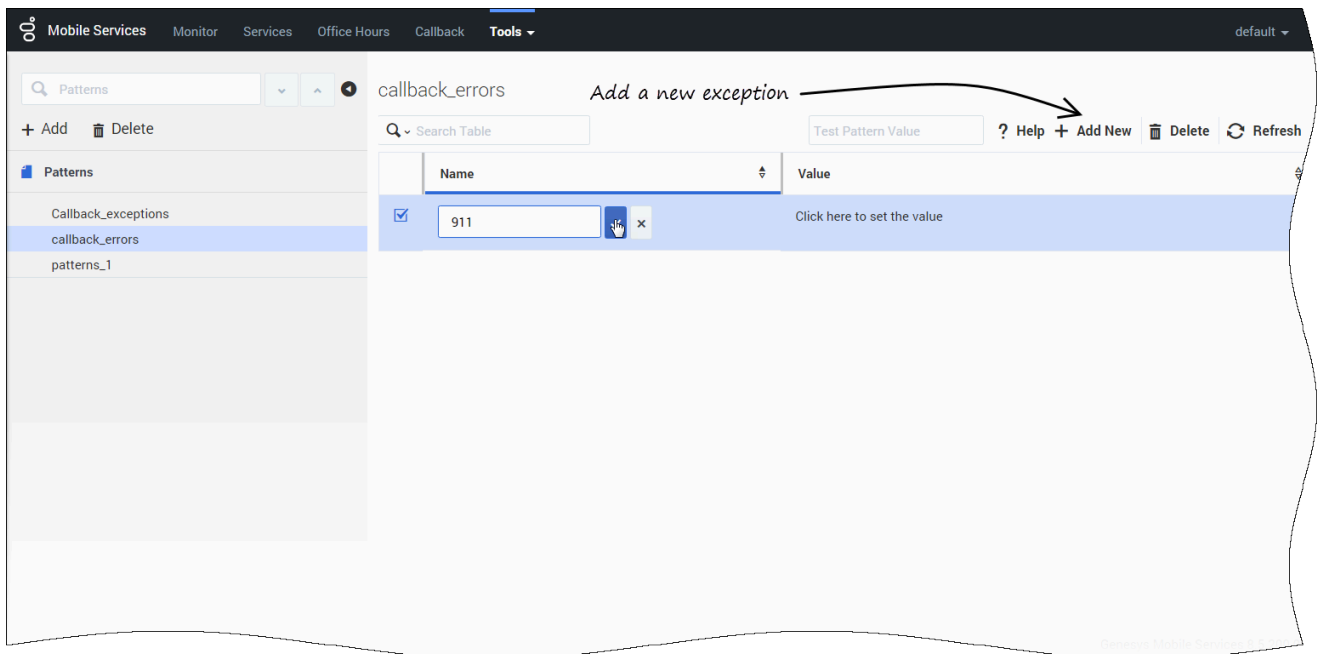
Create a Pattern Group



Exception Patterns

Navigate to **Services and Tools > Tools > Patterns**. Click **Add** to define a new group of patterns. Enter a name.

Create an Exception Pattern



Select a group. Then, in this group, click **Add New** to create as many patterns than you need.

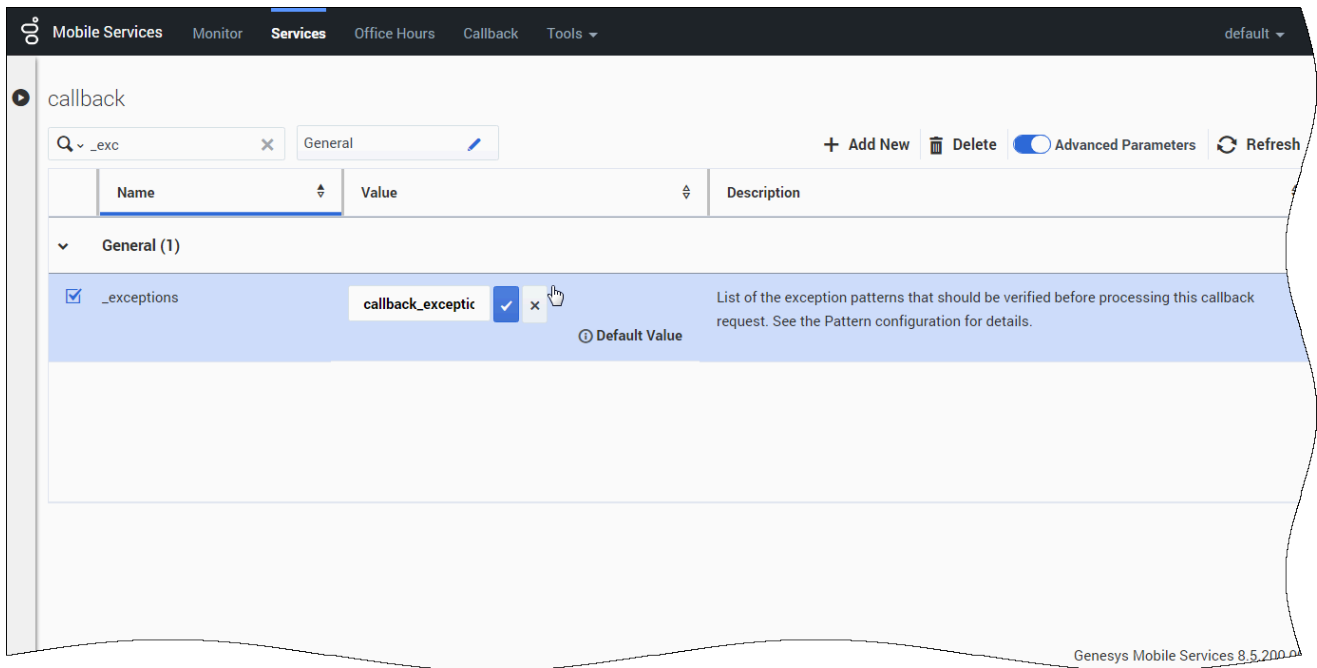
Test your Exception Pattern

The screenshot shows the 'Mobile Services' interface with the 'Tools' menu open. The 'callback_errors' group is selected in the left sidebar. The main area displays a table with two columns: 'Name' and 'Value'. The table contains two rows: '911' and 'bad-emails'. The 'bad-emails' row is highlighted. Above the table, a search bar contains the text 'robot@badrobot.com'. An annotation '1 Enter a string to test your pattern' points to the search bar. Another annotation '2 If any, the matching pattern gets selected' points to the highlighted 'bad-emails' row. The footer of the interface reads 'Genesys Mobile Services 8.5.200.0'.

Name	Value
911	911-
bad-emails	[0-9a-zA-Z]*@badrobot.com

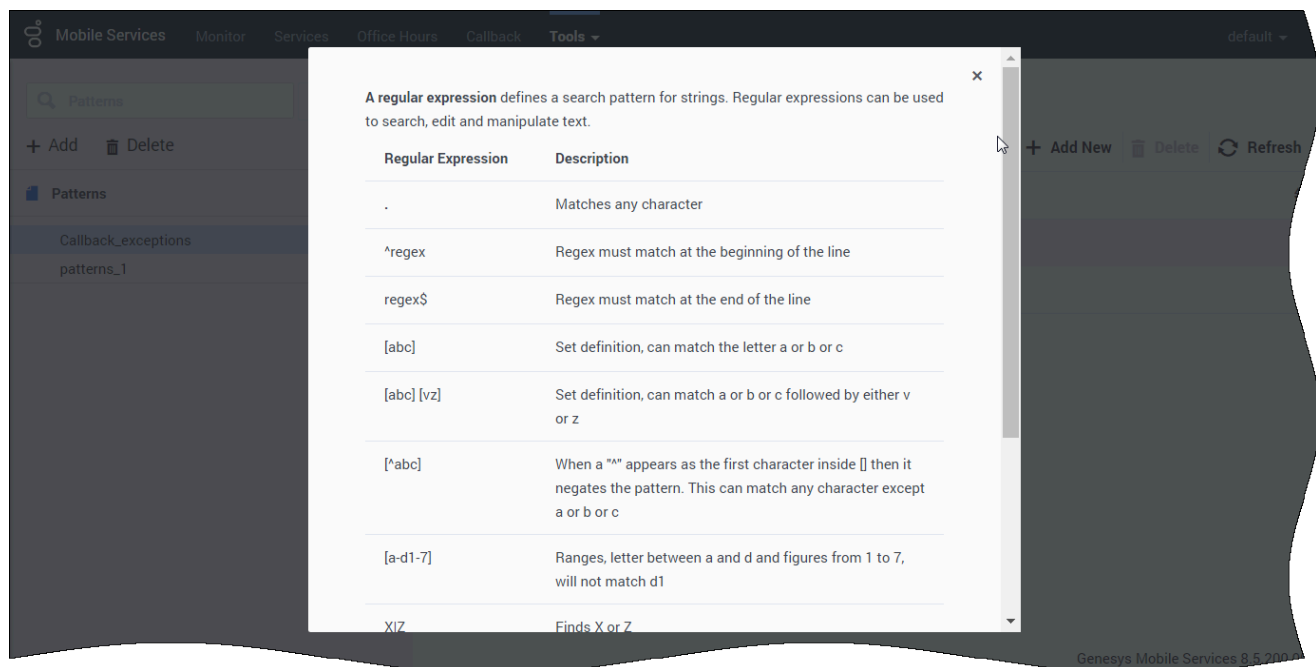
Select a group. You can test the value of a pattern against this entire group. Type a value in the input field, and if a match is found, the corresponding table row will be highlighted.

Add the Pattern Group to your Callback Service



Navigate to **Services and Tools > Services** and expand your Callback service. In the **General** section, set the name of your Pattern Group as the value of the `_exceptions` parameter.

How to get Help about Supported Patterns?



Click the *Help* button to get more information about the supported patterns. For additional details, refer to [Oracle Java Lesson: Regular Expressions](#).

Here are some examples:

```
911=911-  
date=(0[1-9]|1[0-9]|2[0-9]|3[01]).(0[1-9]|1[012]).[0-9]{4}  
date2=(0[1-9]|1[012])[- /.](0[1-9]|1[12][0-9]|3[01])[- /.](19|20)\d\d  
email=^[_A-Za-z0-9-]+(\.[_A-Za-z0-9-]+)*@[A-Za-z0-9-]+(\.[A-Za-z0-9-]+)+$  
bad e-mail address=^[_A-Za-z0-9-]+(\.[_A-Za-z0-9-]+)*@badrobot.com+$  
digit=\\\\d*k*
```