

Webex Connect Email Fails Due To Empty Customer Name Value

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Problem](#)

[Reason for the Failure](#)

[Solution](#)

Introduction

This document describes a scenario where incoming emails fail with "status : 4002 , desc : value is mandatory , name : customer name" error.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Webex Contact Center (WxCC) 2.0
- Webex Connect Portal with Email flows configured

Components Used

The information in this document is based on these software versions:

- WxCC 2.0

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Problem

When the email flow is opened and debug logs are decrypted - the error "**value is mandatory , name : customer name**" is seen at the **Create Task** node.

:mm:ss.ms) : 00 : 00 : 01 . 639

Invoked by : Rule 10945

DETAILS

Node Trans ID : [REDACTED]dec19

Description : status : 4002 , desc : value is mandatory , name : customer name [id : [REDACTED]
a034-0dab7dc6a47a]

Error Description:

"status : 4002 , desc : value is mandatory , name : customer name [id: xxx]â€”"

Reason for the Failure

The problem happens due to an email arrives without a sender name, since **Create Task** node expects the sender name - failure in the flow occurs.

Customer Details

CUSTOMER ID ⓘ

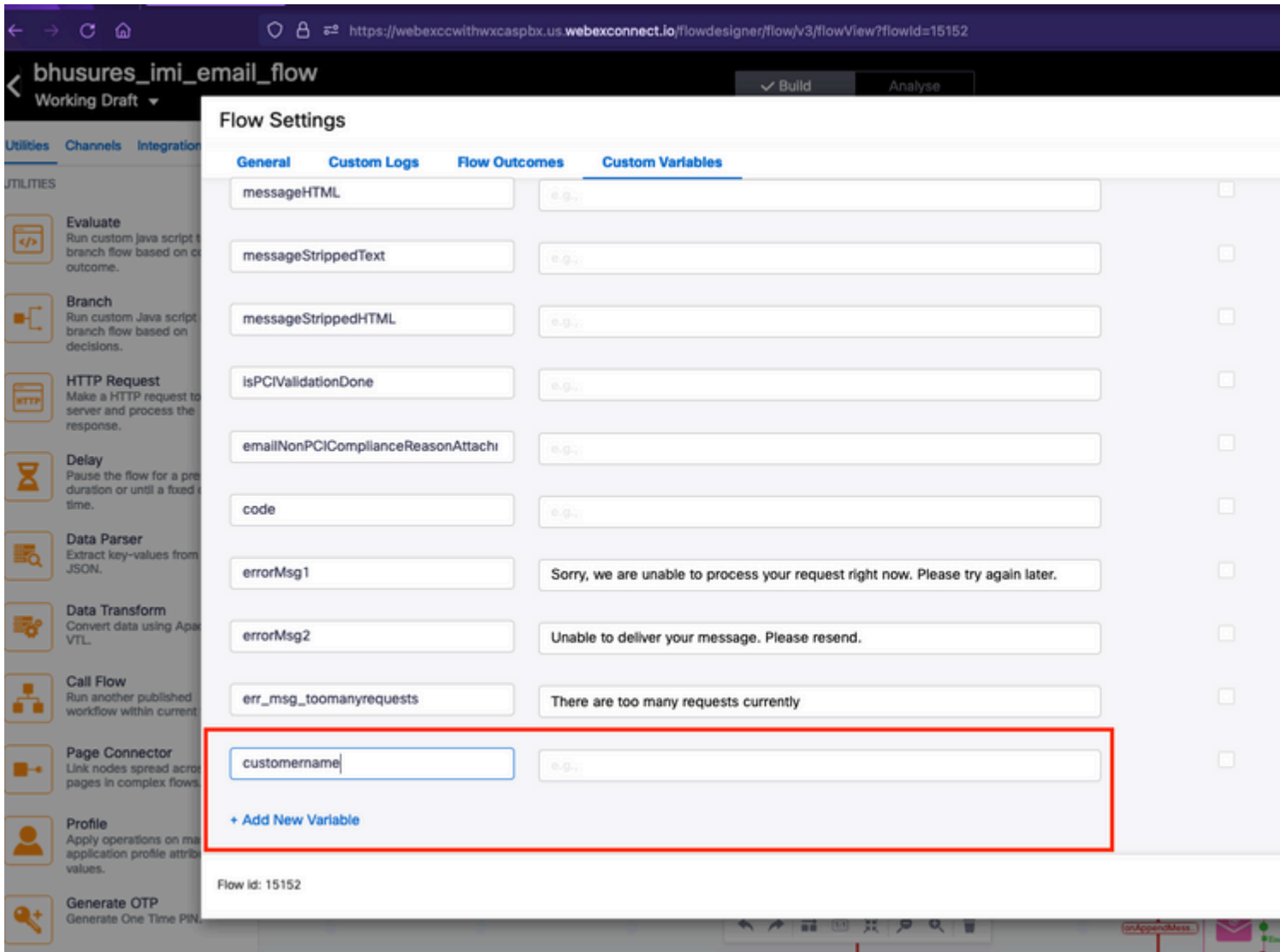
CUSTOMER NAME ⓘ

EMAIL ID (Origin) ⓘ

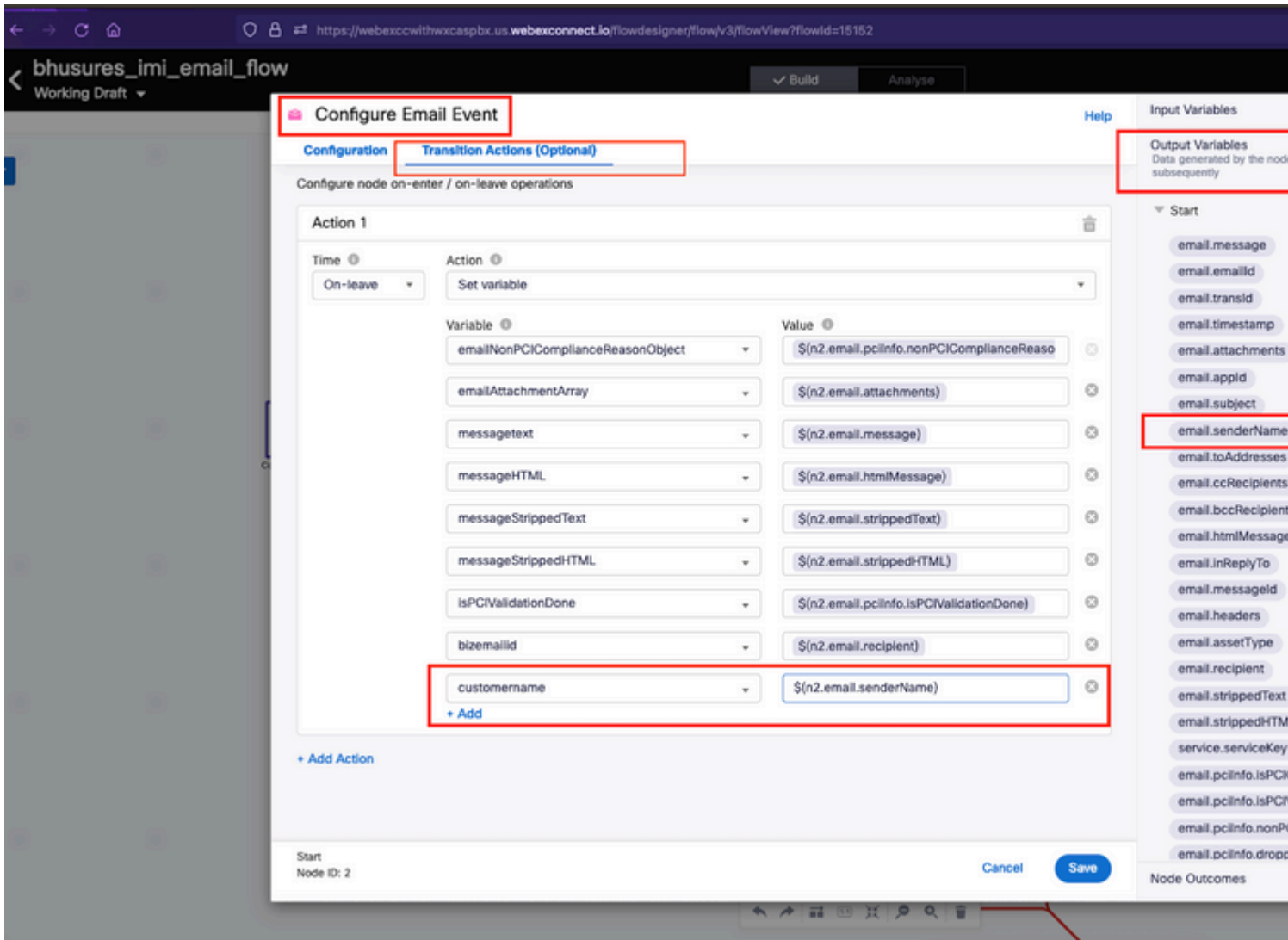
Solution

Modify the email flow to allow a generic from/sender name to be inserted when emails arrive without this mandatory field.

Step 1. Access the email flow and click the gear in the top right corner to access flow settings. Create a new custom variable named "customername"™ and save.



Step 2. Click Configure Email Event Node -> Transition Actions (optional) -> select "add a new variable" -> select the "customername" variable (created in step 1) -> select *email.senderName* under the output variables to add the value -> select Save.



Step 3. Add a generic from/sender name to the parse variables node and save.

In lines 33 and 34 add this condition.

```
if (customername=== '') {  
  customername = 'Generic Sender' ;  
}
```

Parse Variables

Configuration Transition Actions (Optional)

Enter JavaScript to perform logic and return values as output. Configured script outputs will be available as node outcome

```
28 messageBody = "<del>No Content</del>";
29 messageHTML = "<del>No Content</del>";
30 messageStrippedText = "<del>No Content</del>";
31 messageStrippedHTML = "<del>No Content</del>";
32 }
33 - if (customername === "") {
34 customername = "Generic Sender" ;
35 }
36 //Parse PCI Compliance attachment payload for Email
37
38 /*
39
40 1. create one custom variable with name 'emailNonPCIComplianceReasonObject' and assign value 'email.pciInfo.nonPCIComp
41 2. create one custom variable with name 'emailAttachmentArray' and assign value 'email.attachments'
42
```

Configure Script Output

Script Output Branch Name

+ Add New

Test Cancel Save

Evaluate Node ID: 9

Input Variables List of variables available

Custom Variables

Start

Output Variables Node Outcomes

Step 4. Add customername variable to **Create Conversation** and **Create Task** nodes.

Create Conversation

Configuration Transition Actions (Optional)

Conversation Details

Channel ⓘ

Email

Business Email Id ⓘ

\$(bizemailid)

From Email ID ⓘ

\$(n2.email.emailid)

Messages (Optional)

Direction ⓘ

Inbound

Timestamp (In UTC) ⓘ

\$(n2.email.timestamp)

Customer Name ⓘ

\$(customername) ❌

From Address ⓘ

\$(n2.email.emailid)

To Recipients ⓘ

\$(modifiedtoAddress)

CC Recipients ⓘ

\$(n2.email.ccRecipients)

BCC Recipients ⓘ

\$(n2.email.bccRecipients)

Subject ⓘ

\$(n2.email.subject)

Email Headers ⓘ

\$(n2.email.headers)

Plain Body ⓘ

\$(messageBody)

HTML Body ⓘ

\$(messageHTML)

Stripped HTML ⓘ

\$(messageStrippedHTML)

Attachments ⓘ

\$(parseDataAttachment)

Input Variables

List of variables available as input for this node

Search

Custom Variables

- bizemailid
- escapedemailmessage
- escapedhtmlmessage
- modifiedInReplyTo
- parsedattachments
- emailNonPCIComplianceReasonO
- emailAttachmentArray
- conversationId
- messageText
- messageHTML
- messageStrippedText
- messageStrippedHTML
- isPCIVValidationDone
- emailNonPCIComplianceReasonA
- code
- errorMsg1
- errorMsg2
- err_msg_toomanyrequests
- modifiedtoAddress
- customername

Output Variables

Create Task

Configuration

Transition Actions (Optional)

Method Name

Create Task

NODE RUNTIME AUTHORIZATION

WxCC Auth

Task Details

TASK ID ⓘ

\$(fId)

CONVERSATION ID ⓘ

\$(conversationId)

Destination ⓘ

\$(n2.email.appId)

MEDIA TYPE ⓘ

Email

MEDIA CHANNEL ⓘ

Email

SUBJECT ⓘ

\$(n2.email.subject)

Customer Details

CUSTOMER ID ⓘ

\$(n2.email.emalId)

CUSTOMER NAME ⓘ

\$(customername) ❌

EMAIL ID (Origin) ⓘ

\$(n2.email.emalId)

Input Variables

List of variables available as input for this node

Q Search

▶ Custom Variables [F2]

▶ Start Node ID:

▶ Evaluate Node ID:

▶ Search Conversation Node ID:

▶ Re-Open Conversation Node ID:

▶ Create Conversation Node ID:

▶ Append Conversation Node ID:

Output Variables