aws

Developer Guide

Amazon Connect

Amazon Connect: Developer Guide

Table of Contents

What is the Amazon Connect Customer Developer Guide?

> **Note**
>
> Amazon Connect now refers to a portfolio of agentic solutions for business functions. The legacy product is now called Amazon Connect Customer, or simply Customer. The legacy name is used interchangeably in this documentation.

The Amazon Connect Developer Guide is for software developers who build applications that integrate with Amazon Connect using its APIs, flow language, and event-driven architecture.

Who this guide is for

This guide is for developers who:

- Build custom contact center solutions using the Amazon Connect APIs
- Create and manage contact flows programmatically using the Connect flow language
- Automate rules and business logic using the rules function language
- Test contact flows using the testing language
- Integrate Amazon Connect with other AWS services and third-party applications

If you are an administrator setting up and managing Amazon Connect through the console, see the Amazon Connect Administrator Guide.

What's in this guide

Programming with the Amazon Connect Customer APIs

Best practices for calling the APIs, including throttling guidance, SDK support, and eventual consistency behavior.

Actions by resource

A navigation reference that organizes API operations by the resource they act on.

Connect flow language

The JSON-based language for defining contact flows programmatically, including action types, parameters, transitions, and conditions.

Rules function language

The syntax for defining rule conditions and event triggers that automate actions in your contact center.

Testing language

The schema for writing automated tests against your contact flows, including assertions, overrides, and simulated events.

How to use this guide

To understand best practices for integrating with the APIs, start with Programming with the Connect Customer APIs. To build or modify flows programmatically, refer to Connect Customer Flow language for the JSON schema and action reference. To validate your flows before deploying them to production, use Connect Customer testing language.

For the complete API reference with request/response syntax and data types, see the Amazon Connect API Reference.

Programming with the Connect Customer APIs

The following sections provide additional information about using the Connect Customer service APIs.

Contents

Best practices for using Connect Customer APIs

This topic provides guidance for using Connect Customer Describe and List APIs so you don't get unexpected 4xx errors for the response. It also explains how to configure your client Read APIs.

Contents

Types of errors

The Connect Customer APIs provide an HTTP interface. HTTP defines ranges of HTTP Status Codes for different types of error responses.

- Client errors are indicated by HTTP Status Code class of 4xx

- Service errors are indicated by HTTP Status Code class of 5xx

In this reference guide, the documentation for each API has an **Errors** section that includes a brief discussion about HTTP status codes. We recommend looking there as part of your investigation when you get an error.

For information about the common errors returned by Connect Customer public APIs, see Common Errors.

Throttling in Connect Customer APIs

Throttling errors in Connect Customer public API(s) are defined by HTTP status code 429. This HTTP status code can be retried by the client based on their requirement.

> **Important**
>
> The throttling limits are defined for each API separately at the AWS account level, not for the individual Connect Customer instance.

To use any API for Connect Customer resources (such as users, queues, and routing profiles), you need the ID/ARN for the Connect Customer instance.

By default, Connect Customer limits the steady-state requests per second (RPS) across all APIs within an AWS account, per Region. It also limits the burst (that is, the maximum bucket size) across all APIs within an AWS account, per Region.

In Connect Customer the burst limit represents the target maximum number of concurrent request submissions that APIs will fulfill before returning *429 Too Many Requests* error responses.

For more information about throttling quotas, see Connect Customer throttling quotas.

How to configure your client Read API(s)

Your client configuration will vary based on number of resources that your API tries to describe/list per second.

In the following Java example, the number of retries is set to 3. This means after your Connect Customer client implementation experiences throttling, it retries for maximum of 3 times. Instead

of retrying immediately and aggressively, the following snippet waits a specified amount of time (between 0 to max of 5 seconds as defined by maxBackoffTime parameter) between tries and uses EqualJitterBackoffStrategy.

```
final class ClientBuilder {

    private static final int NUMBER_OF_RETRIES = 3;

    private static final RetryPolicy RETRY_POLICY =
            RetryPolicy.builder()
                    .numRetries(NUMBER_OF_RETRIES)
                    .retryCondition(RetryCondition.defaultRetryCondition())
                    .backoffStrategy(EqualJitterBackoffStrategy.builder()
                            .baseDelay(Duration.ofSeconds(1))
                            .maxBackoffTime(Duration.ofSeconds(5))
                            .build())
                    .build();

    public static ConnectClient getClient() {
        return ConnectClient.builder()
                .httpClient(LambdaWrapper.HTTP_CLIENT)

 .overrideConfiguration(ClientOverrideConfiguration.builder().retryPolicy(RETRY_POLICY).build()
                .build();
    }
}
```

When failures are caused by overload or contention, backing off often doesn't help as much as it seems like it should. This is because there's a correlation between failures and backing off/contention:

- If all the failed calls back off to the same time, they cause contention or overload again when they are retried.

To address this, we recommend adding jitter. Jitter adds some amount of randomness to the backoff which spreads the retries around in time. For more information about how much jitter to add and the best ways to add it, see this AWS blog post: Exponential Backoff and Jitter.

For information about types of backoff strategies, see Interface BackoffStrategy.

How to make 2 TPS work for List APIs when you have a large number of resources

There are two options: use List APIs with `maxResults` = 1,000, or use Search APIs as an alternative to List/Describe round trips. Both options are discussed here.

The List API of a particular Connect Customer resource supports a `maxResults` parameter as part of request body. List API(s) support a maximum of 1,000 results in single API call unless specified otherwise in the documentation.

The following example shows the `maxResults` of the ListUsers API.

```
String nextToken = null;
do {
    ListUsersRequest listUsersRequest = ListUsersRequest.builder()
            .instanceId(your Connect Customer instanceId)
            .maxResults(1000)
            .nextToken(nextToken)
            .build();
    ListUsersResponse response = client.listUsers(listUsersRequest);
    nextToken = response.nextToken();
    System.out.println(response.sdkHttpResponse().statusCode());
} while (nextToken != null);
```

If `nextToken` is returned, then more results are available. The value of `nextToken` is a unique pagination token for each page. Make the call again using the returned token to retrieve the next page. Keep all other arguments unchanged. Each pagination token expires after 24 hours. Using an expired pagination token will return an HTTP 400 `InvalidToken` error.

When to use Search APIs instead of List APIs

We recommend you assess the speed of pulling details for 100 records at a time (the Search API limit) instead of pulling 1,000 IDs and doing Describe round trips. It's better to try using Search APIs instead of combination of List and Describe API for a specific resource.

Let's say you have a situation where you're listing specific resources in your Connect Customer instance and then call a Describe API on an individual resource. Instead, we recommend leveraging the Search API for that corresponding resource. Search APIs support several filters that can help to reduce response set as per requirement.

How to make 2 TPS work for Create/Update APIs when you have a large number of resources

There is a performance impact behind creating/updating resources at a default 2 TPS. For example, 100 resources can be created/updated with 2 TPS within 50 seconds. A 1,000 resources with this TPS would need nearly 8 minutes. Based on your use case, if the operation is impacting performance, contact Support and provide a business justification for your request to increase your throttling quota. See How to request an increase to an API throttling quota.

It is your responsibility to always implement the following best practices:

- Check your logic and implement best practices to make your requests as efficient as possible. Check out AWS Well-Architected Tool (AWS WA Tool) for processes that help measure your architecture using AWS best practices.
- Test your requests and any custom processes *before adding them to production operations*.

Hitting a resource quota? Delete unused / stale resources

If you keep hitting the quota limit for a specific resource, we recommend deleting any unused or stale resources. You can find the Delete API for a resource on the resource-specific Action pages. These pages list all the APIs for a given resource.

How to request an increase to an API throttling quota

Important

- We analyze all requests for quota increases and provide guidance for all queries.
- We rarely approve requests if they apply to situations other than those listed below.
- For smaller increase requests, we can approve in hours. Larger increase requests take time to review, process, approve, and deploy. Depending on your specific implementation, your resource, and the size of quota that you want, a request can take up to 3 weeks. An extra-large worldwide increase can potentially take months. If you're increasing your quotas as part of a larger project, keep this information in mind and plan accordingly.

For instructions about how to use the Service Quotas console, see Using the AWS Management Console to request an increase.

In the Services Quotas console, open an Support case and provide the following information:

1. Have you implemented the best practices explained in the Retry behavior topic of the *AWS SDKs and Tools Reference Guide*?
2. What is the performance impact without the requested limit increase? Provide some calculations.
3. What is the expected number of resources customer is trying to create/update/describe every second with the APIs?
4. What is the new quota for the API that you want?

Include in your case if the following situation(s) apply:

- It is a migration request and you need high TPS for a specific time range to configure your instance(s).
- There are performance or business impacting usecases, such as handling huge call volume for peak season.
- You have thousands of resources with multiple concurrent agents working at the same time which might increase the overall traffic from your contact center.

Supported SDKs for all Connect Customer APIs

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Working with Attachments

This topic explains how to work with attachments for Chat and Cases in Connect Customer. For Chat Attachments, refer to the first section on the Connect Customer Participant Service API. For Case Attachments, refer to the second section on the Connect Customer service's Attached Files APIs.

Uploading Attachments with the Participant Service

There are three basic steps for uploading a file using the Connect Customer Participant Service API.

1. HTTP POST file metadata to `StartAttachmentUpload` API, which will provide a signed Amazon S3 URL and attachment ID for uploading the file directly to Amazon S3.
2. HTTP PUT file data to the signed Amazon S3 URL.
3. HTTP POST attachment ID to `CompleteAttachmentUpload` to finalize the upload to Amazon S3.

Below is a basic JavaScript implementation for reference purposes.

```
//Define the html element for file using input tag

<input type="file" id="fileUpload">
<input type="button" id="btnUploadFile" onclick="uploadFile()" value="Upload file">


async function uploadFile() {
    //Initiate the file upload by calling StartAttachmentUpload, providing the name,
 size, and content type of the file you want to upload.
    //The response will include a pre-signed URL and headers to use when building the
 file upload request, as well as an AttachmentId.
    const files = document.getElementById('fileUpload').files;
    const file = files[0];

    const startUploadRequest = {
        AttachmentName: file.name,
        AttachmentSizeInBytes: file.size,
        ContentType: file.type
    };
    const { AttachmentId, UploadMetadata } = await
 startAttachmentUpload(startUploadRequest);
```

```
    //Send the file data to the pre-signed S3 URL.
    //The file is stored in a temporary location in the S3 bucket.

    await uploadFileToS3(file, UploadMetadata.Url, UploadMetadata.HeadersToInclude);

    //Finalize the file upload by calling CompleteAttachmentUpload, providing the
 AttachmentId.
    //This moves the file to the final Attachments S3 path configured for the connect
 instance.
    const completeUploadRequest = {
        AttachmentIds: [ AttachmentId ]
    };
    await completeAttachmentUpload(completeUploadRequest);
}

async function startAttachmentUpload(requestData){
    const response = await fetch(StartAttachmentUploadEndpoint, {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
            'X-Amz-Bearer': ConnectionToken,
        },
        body: JSON.stringify(requestData)
    });

    return await response.json();
}

async function uploadFileToS3(file, signedUrl, headersToInclude) {
    return fetch(signedUrl, {
      method: 'PUT',
      headers: headersToInclude,
      body: file
    });
}

async function completeAttachmentUpload(requestData){
    return fetch(CompleteAttachmentUploadEndpoint, {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
            'X-Amz-Bearer': ConnectionToken,
        },
```

```
        body: JSON.stringify(requestData)
    });
}
```

For an example of a real world implementation, view the Connect Customer ChatJS JavaScript client on GitHub.

Uploading Attached Files with the Connect Service

The same basic steps follow from the Participant service instructions, for usage with the Connect Customer service's attached file APIs.

1. Call the `StartAttachedFileUpload` API with the required attached file information, which will provide a signed Amazon S3 URL, headers, and file ID for uploading the file directly to Amazon S3.

   - If you are attaching a file to a Case, the calling identity must also have the `cases:CreateRelatedItem` permission on the target Case resource.

2. Using your http client of choice, PUT file data to the signed Amazon S3 URL, ensuring that you add the headers as required from the response of Step 1.
3. Call the `CompleteAttachedFileUpload` to finalize the upload to Amazon S3.

Best practices for using PutDialRequestBatch for outbound campaign calling

This topic provides guidance and best practices for using the PutDialRequestBatch API for outbound campaign calling. PutDialRequestBatch takes in a list of DialRequest to be dialed as part of an outbound campaign.

- Before using PutDialRequestBatch, always use the StartCampaign API to start the outbound campaign.
- In DialRequest, set `expirationTime`—the timestamp when a dial request expires—to a future time, preferably a few minutes in the future to allow the outbound campaigns dialer algorithm to attempt to dial.

  Note the following failure codes:

- `InvalidInput`: There's a request input mismatch with the parameters defined in DialRequest. Fix the request input and try again.
- `RequestThrottled`: There aren't enough agents or telecom capacity available. The service maintains an upper limit on the number of records it can accept. When this limit is reached, the service returns a `RequestThrottled` message. This is a signal for you to stop sending more records and instead wait before attempting to retry, allowing the service to process and clear some records so it has room to accept more.
- `UnknownError`: There is an internal failure in outbound campaigns. Please try again.

Recommendations for tracking contact dispositions

A contact disposition is a type of state for a contact. It is either assigned by an agent from a pre-defined list that you provide (for example, **Sold**, **Wrong person**, **Right person, didn't sell**) or assigned by Connect Customer for a contact (for example, `EXPIRED`, `CONTACT_FLOW_DISCONNECT`, or `TELECOM_PROBLEM`). For outbound campaigns, the contact disposition is assigned by Connect Customer.

You can use Connect Customer contact records or contact events to track the contact dispositions for outbound campaigns. The key attribute to track contact disposition is DisconnectReason, which indicates how the voice contact was terminated.

Disconnect reasons can be grouped into the following 3 categories:

- **Success**: A contact is successfully dialed out. Values: `CUSTOMER_DISCONNECT | AGENT_DISCONNECT | THIRD_PARTY_DISCONNECT | BARGED | CONTACT_FLOW_DISCONNECT | OTHER`
- **Expired**: This is expected behavior of the dialing algorithm. When the dialing algorithm determines insufficient dialing capacity (for example, because all agents are occupied or all telecom capacity is being used) during the duration set by the `expirationTime` parameter, Connect Customer does not attempt to call such endpoints. In other words, those endpoints are expired. If you need higher throughput, you can submit a request for higher dialing capacity. Values: `EXPIRED`
- **Failed**: Outbound campaigns failed a dial attempt because there are specific systematic errors. Values: `OUTBOUND_DESTINATION_ENDPOINT_ERROR | OUTBOUND_RESOURCE_ERROR | OUTBOUND_ATTEMPT_FAILED | TELECOM_PROBLEM`

For a description of each of these values, see DisconnectReason in the Contact records data model topic in the *Connect Customer Administrator Guide*.

If the disconnect reason is expired or failed type, take one of the following next steps:

Disconnect reason value	Recommendation
`EXPIRED`	We recommend that you use PutDialRequestBatch to create a new contact for this destination with a different client token.
`OUTBOUND_DESTINATION_ENDPOINT_ERROR`	Check whether the current configuration of your Connect Customer instance allows this destination to be dialed. For more information, see Countries you can call by default.
`OUTBOUND_RESOURCE_ERROR`	• Check whether your Connect Customer instance allows outbound calls. For more information, see Enable outbound calls. • Check that the following resources exist: queue, flow, Connect Customer source phone number, and campaign. • After fixing the configuration of the instance, use PutDialRequestBatch to create a new contact for this destination with a different client token.
`OUTBOUND_ATTEMPT_FAILED`	There is an unidentified error. We recommend that you use PutDialRequestBatch to create a new contact for this destination with a different client token.
`TELECOM_PROBLEM`	Disconnected due to an issue with connecting the call from the carrier, network congestion, network error, etc. We recommend that you use PutDialRequestBatch to create a new contact for this destination with a different client token.

Connect Customer eventual consistency

The Connect Customer API follows an eventual consistency model due to the distributed nature of the system. As a result, changes to Connect Customer resources might not be immediately visible to the subsequent commands you run.

When you perform Connect Customer API calls, there might be a brief delay before the change is available throughout Connect Customer. It typically takes less than a few seconds for the change to propagate throughout the system, but in some cases it can take several minutes. You might get unexpected errors, such as a `ResourceNotFoundException` or an `InvalidRequestException`, during this time. For example, Connect Customer might return a `ResourceNotFoundException` if you call DescribeUser immediately after calling CreateUser.

We recommend that you configure a retry strategy on your Connect Customer clients to automatically retry operations after a brief waiting period. For more information, see Retry behavior in the *AWS SDKs and Tools Reference Guide*.

Resource integrations

CloudFormation

Connect Customer is integrated with CloudFormation, a service that allows you to treat infrastructure as code. Use CloudFormation to model, provision, and manage AWS and third-party resources.

The following Connect Customer resource APIs support CloudFormation templates:

- AWS::Connect::ApprovedOrigin
- AWS::Connect::ContactFlow
- AWS::Connect::ContactFlowModule
- AWS::Connect::EvaluationForm
- AWS::Connect::HoursOfOperation
- AWS::Connect::Instance
- AWS::Connect::InstanceStorageConfig
- AWS::Connect::IntegrationAssociation
- AWS::Connect::PhoneNumber

- AWS::Connect::Prompt
- AWS::Connect::Queue
- AWS::Connect::QuickConnect
- AWS::Connect::RoutingProfile
- AWS::Connect::Rule
- AWS::Connect::SecurityKey
- AWS::Connect::TaskTemplate
- AWS::Connect::TrafficDistributionGroup
- AWS::Connect::User
- AWS::Connect::UserHierarchyGroup
- AWS::Connect::View
- AWS::Connect::ViewVersion

CloudTrail

Connect Customer is integrated with AWS CloudTrail, a service that provides a record of the Connect Customer API calls that a user, role, or AWS service makes. CloudTrail captures Connect Customer API calls as events. All public Connect Customer APIs support CloudTrail.

For more information, see Logging Connect Customer API calls with AWS CloudTrail.

EventBridge

Connect Customer is integrated with Amazon EventBridge, a service that provides a record of the Connect Customer API calls that a user, role, or AWS service makes. All public Connect Customer APIs support EventBridge, with events published to CloudTrail consumable in EventBridge.

Some Connect Customer resources are integrated directly into EventBridge. For more information, see EventBridge events emitted by Connect Customer.

Connect Customer actions by resource

The following section lists the Connect Customer API actions by resource.

- analyticsdataset-api
- agent-status-api
- chat-api
- contacts-api
- data-tables-api
- email-api
- evaluation-api
- files-api
- flows-api
- hierarchy-groups-api
- hours-of-operation-api
- instances-api
- integration-association-api
- metrics-api
- notifications-api
- phone-numbers-api
- prompts-api
- queues-api
- quick-connects-api
- routing-profiles-api
- rules-api
- security-profiles-api
- tags-api
- tasks-api
- trafficdistributiongroups-api
- use-cases-api

- users-api
- view-api
- vocabularies-api
- voice-api
- workspaces-api

Analytics data lake actions

These APIs are in preview release for Connect Customer and are subject to change.

The following API actions are available for analytics datasets:

- AssociateAnalyticsDataSet
- BatchAssociateAnalyticsDataSet
- BatchDisassociateAnalyticsDataSet
- DisassociateAnalyticsDataSet
- ListAnalyticsDataAssociations

Agent status actions

The following API actions are available for agent status:

- CreateAgentStatus
- DescribeAgentStatus
- ListAgentStatuses
- PutUserStatus
- SearchAgentStatuses
- UpdateAgentStatus

> **Note**
>
> Agent status ARNs cannot be deleted but can be renamed or disabled if needed.

For more information about agent status, see About agent status in the *Connect Customer Administrator Guide*.

Chat actions

The following API actions are available for chats:

- CreatePersistentContactAssociation
- SendChatIntegrationEvent
- StartChatContact

For more information about chat, see Concepts: Web and mobile chat capabilities in Connect Customer in the *Connect Customer Administrator Guide*.

Contacts actions

The following API actions are available for contacts:

- AssociateContactWithUser
- BatchPutContact
- CreateContact
- CreateParticipant
- CreatePushNotificationRegistration
- DeletePushNotificationRegistration
- DescribeContact
- DismissUserContact
- GetContactAttributes
- ListAssociatedContacts
- ListContactReferences
- ListRealtimeContactAnalysisSegments
- ListRealtimeContactAnalysisSegmentsV2
- MonitorContact
- SearchContacts

- StartContactStreaming
- StartOutboundChatContact
- StartOutboundVoiceContact
- StartScreenSharing
- StartTaskContact
- StartWebRTCContact
- StopContactStreaming
- StopContact
- TagContact
- TransferContact
- UntagContact
- UpdateContact
- UpdateContactAttributes
- UpdateContactRoutingData
- UpdateContactSchedule

For more information about contacts, see the following topics in the *Connect Customer Administrator Guide*:

- Connect Customer contact events
- Contact records data model
- Capture customer audio: live media streaming
- Customize chat flow experiences by integrating custom participants

Data table actions

The following API actions are available for data tables:

- BatchCreateDataTableValue
- BatchDeleteDataTableValue
- BatchDescribeDataTableValue
- BatchUpdateDataTableValue

- CreateDataTable
- CreateDataTableAttribute
- DeleteDataTable
- DeleteDataTableAttribute
- DescribeDataTable
- DescribeDataTableAttribute
- EvaluateDataTableValues
- ListDataTableAttributes
- ListDataTablePrimaryValues
- ListDataTables
- ListDataTableValues
- SearchDataTables
- UpdateDataTableAttribute
- UpdateDataTableMetadata
- UpdateDataTablePrimaryValues

For more information about data tables, see Data tables in the *Connect Customer Administrator Guide*.

Email actions

The following API actions are available for email:

- AssociateEmailAddressAlias
- CreateEmailAddress
- DeleteEmailAddress
- DescribeEmailAddress
- DisassociateEmailAddressAlias
- SearchEmailAddresses
- SendOutboundEmail
- StartEmailContact

- StartOutboundEmailContact
- UpdateEmailAddressMetadata
- UpdateQueueOutboundEmailConfig

For more information about email, see Set up email in Connect Customer in the *Connect Customer Administrator Guide*.

Evaluation actions

The following API actions are available for evaluations:

- ActivateEvaluationForm
- CreateEvaluationForm
- DeactivateEvaluationForm
- DeleteContactEvaluation
- DescribeContactEvaluation
- DeleteEvaluationForm
- DescribeEvaluationForm
- ListContactEvaluations
- ListEvaluationForms
- ListEvaluationFormVersions
- StartContactEvaluation
- SubmitContactEvaluation
- UpdateContactEvaluation
- UpdateEvaluationForm

To create a CloudFormation template for evaluation forms, see the following topic:

- AWS::Connect::EvaluationForm resource in the *CloudFormation User Guide*

For more information about evaluations, see Evaluate agent performance in the *Connect Customer Administrator Guide*.

Files actions

The following API actions are available for Files:

- StartAttachedFileUpload
- CompleteAttachedFileUpload
- GetAttachedFile
- BatchGetAttachedFileMetadata
- DeleteAttachedFile

For more information about attached files, see Enable attachments to share files using chat and upload files to cases in the *Connect Customer Administrator Guide*.

Flows / Flow module actions

The following API actions are available for flows and flow modules:

- AssociateFlow
- BatchGetFlowAssociation
- CreateContactFlow
- CreateContactFlowModule
- CreateContactFlowVersion
- DeleteContactFlow
- DeleteContactFlowModule
- DescribeContactFlow
- DescribeContactFlowModule
- DisassociateFlow
- GetFlowAssociation
- ListContactFlows
- ListContactFlowModules
- ListContactFlowVersions
- ListFlowAssociations
- SearchContactFlows

- SearchContactFlowModules
- UpdateContactFlowContent
- UpdateContactFlowMetadata
- UpdateContactFlowModuleContent
- UpdateContactFlowModuleMetadata
- UpdateContactFlowName

To create a CloudFormation template for flows, see the following topics:

- AWS::Connect::ContactFlow
- AWS::Connect::ContactFlowModule

Use the Flow language to programmatically write flows rather than drag blocks onto the flow designer.

For more information about flows, see Create Connect Customer flows in the *Connect Customer Administrator Guide*.

Hierarchy group actions

The following API actions are available for hierarchy groups:

- CreateUserHierarchyGroup
- DeleteUserHierarchyGroup
- DescribeUserHierarchyGroup
- DescribeUserHierarchyStructure
- ListUserHierarchyGroups
- SearchUserHierarchyGroups
- UpdateUserHierarchy
- UpdateUserHierarchyGroupName
- UpdateUserHierarchyStructure

To create a CloudFormation template for hierarchy groups, see the following topic:

- AWS::Connect::UserHierarchyGroup

For more information about managing users, see Manage users in Connect Customer in the *Connect Customer Administrator Guide*.

For more information about agent hierarchies, see Set up agent hierarchies in the *Connect Customer Administrator Guide*.

Hours of operation actions

The following API actions are available for managing hours of operation:

- CreateHoursOfOperation
- DescribeHoursOfOperation
- DeleteHoursOfOperation
- ListHoursOfOperations
- SearchHoursOfOperations
- UpdateHoursOfOperation
- ListChildHoursOfOperations
- AssociateHoursOfOperations
- DisassociateHoursOfOperations

The following API actions are available for managing hours of operation overrides:

- CreateHoursOfOperationOverride
- DeleteHoursOfOperationOverride
- DescribeHoursOfOperationOverride
- GetEffectiveHoursOfOperations
- ListHoursOfOperationOverrides
- SearchHoursOfOperationOverrides
- UpdateHoursOfOperationOverride

To create a CloudFormation template for hours of operation, see the following topic:

- AWS::Connect::HoursOfOperation

For more information about hours of operation and overrides, see Set the hours of operation and timezone for a queue in the *Connect Customer Administrator Guide*.

Instance actions

The following API actions are available for instances:

- AssociateApprovedOrigin
- AssociateBot
- AssociateInstanceStorageConfig
- AssociateLambdaFunction
- AssociateLexBot
- AssociateSecurityKey
- CreateInstance
- DeleteInstance
- DescribeInstance
- DescribeInstanceAttribute
- DescribeInstanceStorageConfig
- DisassociateApprovedOrigin
- DisassociateBot
- DisassociateInstanceStorageConfig
- DisassociateLambdaFunction
- DisassociateLexBot
- DisassociateSecurityKey
- GetFederationToken
- ListApprovedOrigins
- ListBots
- ListInstances
- ListInstanceAttributes
- ListInstanceStorageConfigs

- ListLambdaFunctions
- ListLexBots
- ListSecurityKeys
- ReplicateInstance
- UpdateInstanceAttribute
- UpdateInstanceStorageConfig

To create a CloudFormation template for instances, see the following topics:

- AWS::Connect::Instance
- AWS::Connect::InstanceStorageConfig

For more information about instances, see Create an Connect Customer instance in the *Connect Customer Administrator Guide*.

Integration association actions

The following API actions are available for integration associations:

- CreateIntegrationAssociation
- DeleteIntegrationAssociation
- ListIntegrationAssociations

To create a CloudFormation template for instance associations, see the following topic:

- AWS::Connect::IntegrationAssociation

For more information about integration associations, see Set up pre-built integrations in the *Connect Customer Administrator Guide*.

Metrics actions

The following API actions are available for metrics:

- GetContactMetrics

- GetCurrentMetricData
- GetCurrentUserData
- GetMetricData
- GetMetricDataV2

For more information about metrics, see Metrics, dashboards, and reports in Connect Customer in the *Connect Customer Administrator Guide*.

Notifications actions

The following API actions are available for notifications:

- CreateNotification
- DeleteNotification
- DescribeNotification
- ListNotifications
- ListUserNotifications
- SearchNotifications
- UpdateNotificationContent
- UpdateUserNotificationStatus

For more information about notifications, see Notifications Connect Customer in the *Connect Customer Administrator Guide*.

Phone number actions

The following API actions are available for phone numbers:

- AssociatePhoneNumberContactFlow
- ClaimPhoneNumber
- DescribePhoneNumber
- DisassociatePhoneNumberContactFlow
- ImportPhoneNumber

- ListPhoneNumbers
- ListPhoneNumbersV2
- ReleasePhoneNumber
- SearchAvailablePhoneNumbers
- UpdatePhoneNumber
- UpdatePhoneNumberMetadata

To create a CloudFormation template for phone numbers, see the following topic:

- AWS::Connect::PhoneNumber

For more information about phone numbers, see Set up phone numbers for your contact center in the *Connect Customer Administrator Guide*.

Predefined attributes actions

The following API actions are available for predefined attributes:

- CreatePredefinedAttribute
- DescribePredefinedAttribute
- ListPredefinedAttributes
- SearchPredefinedAttributes
- UpdatePredefinedAttribute

To create a CloudFormation template for predefined attributes, see the following topic:

- AWS::Connect::PredefinedAttribute

For more information about predefined attributes, see Create predefined attributes for routing contacts to agents in the *Connect Customer Administrator Guide*.

Prompt actions

The following API actions are available for prompts:

- CreatePrompt
- DeletePrompt
- DescribePrompt
- GetPromptFile
- ListPrompts
- SearchPrompts
- UpdatePrompt

To create a CloudFormation template for prompts, see the following topic:

- AWS::Connect::Prompt

For more information about prompts, see Create prompts in the *Connect Customer Administrator Guide*.

Queue actions

The following API actions are available for queues:

- CreateQueue
- DeleteQueue
- DescribeQueue
- ListQueues
- SearchQueues
- UpdateQueueHoursOfOperation
- UpdateQueueMaxContacts
- UpdateQueueName
- UpdateQueueOutboundCallerConfig
- UpdateQueueStatus

To create a CloudFormation template for queues, see the following topic:

- AWS::Connect::Queue

For more information about queues, see Set up routing in the *Connect Customer Administrator Guide*.

Quick connect actions

The following API actions are available for quick connects:

- AssociateQueueQuickConnects
- CreateQuickConnect
- DeleteQuickConnect
- DescribeQuickConnect
- DisassociateQueueQuickConnects
- ListQueueQuickConnects
- ListQuickConnects
- SearchQuickConnects
- UpdateQuickConnectConfig
- UpdateQuickConnectName

To create a CloudFormation template for quick connects, see the following topic:

- AWS::Connect::QuickConnect

For more information about quick connects, see Create quick connects in the *Connect Customer Administrator Guide*.

Routing profile actions

The following API actions are available for routing profiles:

- AssociateRoutingProfileQueues
- CreateRoutingProfile
- DeleteRoutingProfile
- DescribeRoutingProfile
- DisassociateRoutingProfileQueues

- ListRoutingProfileManualAssignmentQueues
- ListRoutingProfiles
- ListRoutingProfileQueues
- SearchRoutingProfiles
- UpdateRoutingProfileConcurrency
- UpdateRoutingProfileDefaultOutboundQueue
- UpdateRoutingProfileName
- UpdateRoutingProfileQueues

To create a CloudFormation template for routing profiles, see the following topic:

- AWS::Connect::RoutingProfile

For more information about routing profiles, see Create a routing profile in the *Connect Customer Administrator Guide*.

Rules actions

The following API actions are available for rules:

- CreateRule
- DeleteRule
- DescribeRule
- ListRules
- UpdateRule

To create a CloudFormation template for rules, see the following topic:

- AWS::Connect::Rule

Use the Rules Function language to code conditions for the rules.

For more information about Connect Customer rules, see Create rules and Create rules with Contact Lens in the *Connect Customer Administrator Guide*.

Security profile actions

The following API actions are available for security profiles:

- CreateSecurityProfile
- DescribeSecurityProfile
- DeleteSecurityProfile
- ListSecurityProfiles
- ListSecurityProfileApplications
- ListSecurityProfilePermissions
- SearchSecurityProfiles
- UpdateSecurityProfile

To create a CloudFormation template for tasks, see the following topic:

- AWS::Connect::SecurityProfile

For more information about security profiles, see Security profiles in the *Connect Customer Administrator Guide*.

Tag actions

The following API actions are available for tags:

- SearchResourceTags
- TagResource
- ListTagsForResource
- UntagResource

For more information about tags, see Tagging resources in Connect Customer in the *Connect Customer Administrator Guide*.

Task actions

The following API actions are available for tasks:

- CreateTaskTemplate
- DeleteTaskTemplate
- GetTaskTemplate
- ListTaskTemplates
- StartTaskContact
- UpdateTaskTemplate

To create a CloudFormation template for tasks, see the following topic:

- AWS::Connect::TaskTemplate

For more information about tasks, see the following topics in the *Connect Customer Administrator Guide*:

- Tasks
- Set up tasks
- Set up applications for task creation

Traffic distribution group actions

The following API actions are available for traffic distribution groups:

- AssociateTrafficDistributionGroupUser
- CreateTrafficDistributionGroup
- DeleteTrafficDistributionGroup
- DescribeTrafficDistributionGroup
- DisassociateTrafficDistributionGroupUser
- GetTrafficDistribution
- ListTrafficDistributionGroups
- ListTrafficDistributionGroupUsers
- UpdateTrafficDistribution

To create a CloudFormation template for traffic distribution groups, see the following topic:

- AWS::Connect::TrafficDistributionGroup

For more information about traffic distribution groups, see Set up Connect Customer Global Resiliency in the *Connect Customer Administrator Guide*.

Use cases actions

The following API actions are available for use cases:

- CreateUseCase
- DeleteUseCase
- ListUseCases

User management actions

The following API actions are available for user management:

- AssociateUserProficiencies
- CreateUser
- DeleteUser
- DescribeUser
- DisassociateUserProficiencies
- ListUsers
- ListUserProficiencies
- PutUserStatus
- SearchUsers
- UpdateUserConfig
- UpdateUserIdentityInfo
- UpdateUserPhoneConfig
- UpdateUserRoutingProfile
- UpdateUserSecurityProfiles

To create a CloudFormation template for users, see the following topic:

- AWS::Connect::User

For more information about managing users, see Manage users in Connect Customer in the *Connect Customer Administrator Guide*.

View actions

The following API actions are available for views:

- CreateView
- CreateViewVersion
- DeleteView
- DeleteViewVersion
- DescribeView
- ListViews
- ListViewVersions
- UpdateViewContent
- UpdateViewMetadata
- View
- ViewContent
- ViewInputContent
- ViewSummary
- ViewVersionSummary

To create a CloudFormation template for views, see the following topics:

- AWS::Connect::View
- AWS::Connect::ViewVersion

For more information about views, see the following topic in the *Connect Customer Administrator Guide*:

- View resource

Vocabulary actions

The following API actions are available for vocabularies:

- AssociateDefaultVocabulary
- CreateVocabulary
- DeleteVocabulary
- DescribeVocabulary
- ListDefaultVocabularies
- SearchVocabularies

For more information about custom vocabularies, see Add custom vocabularies in the *Connect Customer Administrator Guide*.

Voice actions

The following API actions are available for voice:

- ResumeContactRecording
- StartContactRecording
- StartOutboundVoiceContact
- StopContactRecording
- SuspendContactRecording

For more information, see the following topics in the *Connect Customer Administrator Guide*:

- Monitor live conversations
- Review recorded conversations

Workspace actions

The following API actions are available for workspaces:

- AssociateWorkspace

- CreateWorkspace
- CreateWorkspacePage
- DeleteWorkspace
- DeleteWorkspaceMedia
- DeleteWorkspacePage
- DescribeWorkspace
- DisassociateWorkspace
- ImportWorkspaceMedia
- ListWorkspaceMedia
- ListWorkspacePages
- ListWorkspaces
- SearchWorkspaceAssociations
- SearchWorkspaces
- UpdateWorkspaceMetadata
- UpdateWorkspacePage
- UpdateWorkspaceTheme
- UpdateWorkspaceVisibility

For more information about workspaces, see Workspaces in the *Connect Customer Administrator Guide*.

Connect Customer Flow language

This section describes the Connect Customer Flow language and how to use it. The Flow language is a JSON-based representation of a series of flow actions, and the criteria for moving between them.

We've provided you with the Flow language so you can:

- Efficiently update flows that you're migrating from one instance to another.
- Write flows rather than drag blocks onto the flow designer.

Contents

Connect Customer Flow language concepts

The following terms are used in the Flow language.

Contact

Flows can be run in context of a contact. In this case, they are referred to as *flows*.

Participant

Flows can additionally be run in a participant context. This allows participant actions—such as playing prompts or getting customer input—to be run. Certain types of flows, such as "No participants remaining" disconnect flows and Workitem flows, don't have a participant associated.

Action types

Flow actions have the following implicit types associated with them. A type determines when an action is attempted.

- Contact actions in the Connect Customer Flow language. These actions are attempted only when the flow is run in context of a contact. They generally result in contact data being manipulated in some way.
- Flow control actions in the Connect Customer Flow language. These actions are used only to determine the path through a flow. They have no side effects. Certain data may not be available. For example, contact data isn't available if the action is determining its path based on contact data. These actions generally work in every circumstance.
- Interactions in the Connect Customer Flow language. These actions have side effects, but don't require a contact or a participant. Interactions include actions such as invoking an AWS Lambda function. They generally work in every circumstance.
- Participant actions in the Connect Customer Flow language. These actions are attempted only when the flow is run in context of a participant. They generally result in an action that the participant experiences, such as playing a prompt or disconnecting.

Example flow in Connect Customer Flow language

The following example shows a JSON string that represents the content of the flow. It shows a simple flow that plays a prompt using static text and disconnects.

To learn how to get block identifiers, we recommend creating a new flow in Connect Customer console, and then calling the DescribeContactFlow API for it.

```
{
    "Version": "2019-10-30",  //A string representing the version of the Flow.
Currently the only supported version is 2019-10-30.

    "StartAction": "12345678-1234-1234-1234-123456789012", //A string representing the
first Action to run when the flow starts running.
                                                            //In this case, it's the
identifier of the Play prompt block.
                                                           //The value of this field must
match the Identifier of an Action in the Actions list.
    "Metadata": { //An object that may be filled in with data as desired.
        "EntryPointPosition": {
            "x": 88,
            "y": 100
        },
        "ActionMetadata": {
```

```
            "12345678-1234-1234-1234-123456789012": {//The identifier of the Play
 prompt block.
                "Position": {
                    "x": 270,
                    "y": 98
                }
            },
            "abcdef-abcd-abcd-abcd-abcdefghijkl": {//The identifier of the Disconnect/
hang up block.
                "Position": {
                    "x": 545,
                    "y": 92
                }
            }

        }
    },
    "Actions": [  //A list of individual Action objects. These Actions are the
 definition of the Flow's behavior and are detailed below.
                  //A single Flow may have no more than 250 Actions defined.
        {
            "Identifier": "12345678-1234-1234-1234-123456789012", //The identifier of
 the Play prompt block.
            "Type": "MessageParticipant",  //This is the flow action.
            "Transitions": {
                "NextAction": "abcdef-abcd-abcd-abcd-abcdefghijkl", //The identifier of
 the Disconnect/hang up block.
                "Errors": [],
                "Conditions": []
            },
            "Parameters": {
                "Text": "Thanks for calling the sample flow!"
            }
        },
        {
            "Identifier": "abcdef-abcd-abcd-abcd-abcdefghijkl",  //The identifier of
 the Disconnect/hang up block.
            "Type": "DisconnectParticipant",  //This is the flow action.
            "Transitions": {},
            "Parameters": {}
        }
    ]
}
```

Actions in the Connect Customer Flow Language

An Action is a single step of a flow's run. This topic describes the fields that must be defined.

Identifier

A string that must be unique among all Actions within the same Flow. This Identifier can be up to 50 characters long, and can include any characters (including unicode and spaces). They can be opaque or user-friendly.

Important

- The following characters are not allowed in the Identifier field: (% : (\ /) = $, ; [ ] { })
- The following strings are not allowed in the Identifier field: __proto__, constructor, __defineGetter__, __defineSetter__, toString, hasOwnProperty, isPrototypeOf, propertyIsEnumerable, toLocaleString, and valueOf.

Type

A string that identifies the type of action being performed for a particular step of the Flow. This type must be one of a list of allowable Types, which are covered later.

Parameters

An object that defines the customizable behavior of a particular Action block. Each Action has its own format of this Parameters object, which is detailed in the individual Actions definition.

The Parameters object defines customizable behavior for the Action. For example, it defines which Attributes to set or which AWS Lambda function to run. The format differs for each Action type. To find the specific format of a specific Action's Parameter object, see the individual Action's definition below.

Transitions

An object that defines the behavior for choosing the next Action after the current Action completes. Certain Actions terminate, meaning that they finish running the flow when they're run. This is because Transitions must be defined as an empty object.

The Transitions object defines how to proceed to the next Action during flow runtime. This object must have the following fields specified:

NextAction

NextAction is a string that contains the Identifier of the Action that should be run after this Action, if no error or condition is preferentially chosen.

Errors

Errors is a list of error objects. Each error object contains a type or category of error (ErrorType), and the Identifier of the Action that should be run subsequently when that error occurs (NextAction).

Each individual Action supports specific Errors. These are detailed in the Action's definition.

Conditions

Conditions are an ordered list that defines a series of checks to evaluate against the Action's result. This result changes per Action and can also change based on Parameters - examples of these are "the number of contacts in queue" for the CheckMetricData Action if the MetricType parameter refers to the NumberOfContactsInQueue, and "the value of the attribute" for the Compare Action. Conditions are evaluated in order, and the first Condition that evaluates to true will result in it being chosen as the Transition to occur, making that Condition's Target the next Action run. The Conditions object is explained in more detail below.

A Condition is a definition of how to evaluate an Action's result, and may evaluate to true or false. The Conditions object on the flow contains an ordered list of objects. Each object contains a NextAction (the Identifier of the Action to be invoked if the Condition evaluates to be true) and the Condition to evaluate:

- NextAction: A string that contains the Identifier of the Action that should be run after this Action if this Condition is the first condition to evaluate to true.
- Condition: An object that defines the evaluation logic.

The Condition object

The Condition object must contain the following fields:

- **Operator**: A string that indicates which comparison operator that is applied to the Operands. The list of allowed Operators and a description of their logic is defined in the following table.
- **Operands**: A list of operands to which the Operator is applied. Depending on the Operator, these Operands may be strings or they may be Condition objects. The specific Operator defines which type of Operand is expected, along with the number of Operands expected (some Operators will require only one Operand, some will support a list of up to ten Operands). Conditions may be nested no more than five Conditions deep, and a single Condition may not contain more than 50 sub-Conditions, regardless of how deeply nested they are.

List of Operators

Operator	Description	Operand type	Operand count
Equals	Returns **true** if the string specified exactly equals the result.	String	One
TextStartsWith	Returns **true** if the result, interpreted as text, begins with the specified string.	String	One
TextEndsWith	Returns **true** if the result, interpreted as text, ends with the specified string.	String	One
TextContains	Returns **true** if the result, interpreted as text, contains the specified string at least once.	String	One

Operator	Description	Operand type	Operand count
NumberGreaterThan	Returns **true** if the result, interpreted as a numeric value, is larger than the specified string. If either the result or the specified string are not numeric, returns **false**.	String	One
NumberGreaterOrEqualTo	Returns **true** if the result, interpreted as a numeric value, is larger than or equal to the specified string. If either the result or the specified string are not numeric, returns **false**.	String	One
NumberLessThan	Returns **true** if the result, interpreted as a numeric value, is smaller than the specified string. If either the result or the specified string are not numeric, returns **false**.	String	One

Operator	Description	Operand type	Operand count
NumberLessOrEqualTo	Returns **true** if the result, interpreted as a numeric value, is smaller than or equal to the specified string. If either the result or the specified string are not numeric, returns **false**.	String	One

Example Condition

Following is an example of a condition that returns true if the result starts with "ABC":

```
{
    "Operator": "TextStartsWith",
    "Operands": [
        "ABC"
    ]
}
```

Parameter restrictions for actions in the Connect Customer Flow language

There are several restrictions on parameters. Here's what they mean:

- Must be defined statically. This means that JSONPath cannot be used at all in this value.
- Must be defined statically or as a single valid JSONPath identifier.

  If JSONPath is used, it must be the entirety of the value; you can't specify an input of "My name is $.Name". Further, the JSONPath must be valid - $.Attributes.stuff is okay, $.BadValue is not okay because there's no "BadValue" path on the object used by flows.
- May be defined statically or dynamically. Anything goes. A value of "My name is $.Name" is fine here, as well as a fully static value.

Contact actions in the Connect Customer Flow language

Contact actions are attempted only for flows that run in context of a contact. They generally result in contact data being manipulated in some way.

Contents

- UpdateContactTextToSpeechVoice
- UpdatePreviousContactParticipantState

CompleteOutboundCall

When a flow is run before an outbound call is made as part of an outbound contact, this action calls the outbound destination. If this action is not used, the first participant action implicitly completes the outbound call.

Parameter object

```
""CallerId": { Optional, an override of the caller ID to present when calling. Is
 ignored when using VoiceConnectors
  "Number": The caller ID number to present when calling. Can either be fully static or
 a single valid JSONPath identifier
}
"VoiceConnector": { Optional, Configuration of the voice connector
   "VoiceConnectorType": Only support "ChimeConnector". Must be defined statically.
   "VoiceConnectorArn": The ARN of the Voice Connector. Can be set statically or
 dynamically.
   "FromUser": The user who makes the call. Can be set statically or dynamically.
   "ToUser": The user who receives the call. Can be set statically or dynamically.
   "UserToUserInformation": Optional, SIP user-to-user information. Can be set
 statically or dynamically.
}
"ConnectionTimeLimitSeconds": Optional, Only used for Voice Connector use case. An
 integer between 1 and 600
                                                          (inclusive) representing the
 number of seconds
                                                        to wait for the voice connector
 to answer before
                                                        canceling the call. Can be set
 statically or dynamically.
```

Results and conditions

None.

Errors

None.

Restrictions

This action can be used only when the contact is in the process of making an outbound call, but has not yet called the outbound number.

Corresponding block in the UI

Call phone number

CreateCase

Creates a new case using an existing case template. Templates come with predefined fields, some of which are required and will appear in a side panel when you start. You can review and fill in the required fields, and optionally update any other available fields based on your needs. These fields are set up in your instance ahead of time. You can also choose to link a contact to the new case if needed.

Parameter object

```
"Parameters": {
        "LinkContactToCase": "true" or "false", If set to true, cases will open
 automatically when the agent accepts the contact.
        "CaseTemplateId": A templateId aligned with the existing case templateName
        "CaseRequestFields": An optional map of case fields to be set. Keys should be
 fields from Cases domain. Values can be static or dynamic.
}
```

Results and conditions

None.

Errors

- ContactNotLinked - If you specify to link the contact to case, then this error branch will appear. It might be that the contact was not linked after the case is retrieved (partial success/partial failure). If this happens, then the flow will follow this branch.
- NoMatchingError - An error was encountered while trying to find the case. This may be due to a system error or how CreateCase is configured.

Restrictions

None. This can be used in any type of flow and any channel.

Corresponding block in the UI

Flow block: Cases

CreateTask

Creates a new task to run an assigned flow.

Parameter object

```
"Parameters":
{
        "ContactFlowId": A flow ID or flow ARN. *Must be fully static or a single valid
 JSONPath identifier*,
       "Attributes": [Optional] { an Object that holds the attributes to be set.
         "Key": "Value" Both the key and value may be defined statically or
 dynamically.
       },
       "Name": Name of the task that needs to be created. This is a string.,
       "Description": [Optional] Description of the task that needs to be created.
 This is a string.,
       "References": [Optional] { an Object that holds the references to be set.
         "Type": "Value" Both the key and value may be defined statically or
 dynamically.
       },
       "DelaySeconds": [Optional] Time in seconds after which the task should be
 created. This is used to schedule the task by the agent.
       The integer value between 1 and 518400 (6 days). If ScheduledTime is specified,
 this parameter may not be specified.
       "ScheduledTime": [Optional] The date and time at which the task should be
 created. If DelaySeconds is specified, this parameter may not be specified.
       "TaskTemplateId": [Optional] ID of the task template that will be used to
 create the task. This must be defined statically.
     }
}
```

Results and conditions

None. No conditions are supported.

Errors

NoMatchingError - if no other Error matches.

Restrictions

This action is supported on all channels and in all contact flow types.

Corresponding block in the UI

Flow block: Create task

CreateWisdomSession

Associates a Wisdom domain to a contact that is being executed in a Flow to enable real-time recommendations on the current contact.

Parameter object

```
{
    "WisdomAssistantArn":  ARN for the Wisdom Assistant. May be specified statically or
 dynamically.
}
```

Results and conditions

None. No conditions are supported.

Errors

NoMatchingError - if no other Error matches.

Restrictions

This action is only supported on the voice channel. This action can be used in all contact flow types.

Corresponding block in the UI

Flow block: Amazon Q in Connect

DequeueContactAndTransferToQueue

This action is a combination of a "Dequeue" action and a "TransferContactToQueue" action. This means that a contact in a queue is removed from the queue, a new contact segment is created with the existing contact as its previous contact, and the new contact is placed into the specified queue (referred to as "Queue-to-queue transfer"). If this contact has not been queued, is actively being joined to an agent, or has been routed to an agent, this action fails.

Parameter object

```
{
    "QueueId": [Optional] A queue ID or queue ARN. If AgentId is specified, this may
 not be specified. Must be either fully statically defined or a single, valid JSONPath
 identifier.
    "AgentId": [Optional] An agent ID or agent ARN, representing an agent queue. If
 QueueId is specified, this may not be specified. Must be either fully statically
 defined or a single, valid JSONPath identifier.
}
```

Results and conditions

None.

Errors

- QueueAtCapacity - if the destination queue is at capacity and the contact cannot be queued within it.
- NoMatchingError - if no other Error matches.

Restrictions

This action is only supported in the customer queue flow. It is not supported in any other type of flow.

Corresponding block in the UI

Maps to Transfer to queue block but only when used in a Customer queue flow.

EndFlowModuleExecution

Ends the current module execution without disconnecting the contact.

Parameter object

```
{

}
```

Results and conditions

None. No conditions are supported.

Errors

None

Restrictions

This action is available only in flow modules.

Corresponding block in the UI

Return Block (not yet documented)

GetCase

Searches all existing cases with the provided customer ID. Add request fields to filter by case fields. Specify the case fields to be returned in the response to persist in the context.

Parameter object

```
{
    "LinkContactToCase": "true" or "false", If set to true, cases will open
 automatically when the agent accepts the contact.
   "GetLastUpdatedCase": "true" or "false", You can specify to get only the last
 updated case for any search criteria.
    "CustomerId": "Customer's Id", Search all cases for this customer Id.
    "CaseRequestFields": An optional map of cases request fields. Keys should be fields
 from Cases domain. Values can be static or dynamic.
    "CaseResponseFields": [ ] An optional list of field names that should be persisted
 in the case namespace.
}
```

Results and conditions

None.

Errors

- NoMatchingError - An error was encountered while trying to find the case. This may be due to a system error or how Get case is configured.

- ContactNotLinked - If you specify to link the contact to case, then this error branch will appear. It might be that the contact was not linked after the case is retrieved (partial success/partial failure). If this happens, then the flow will follow this branch.
- MultipleFound - Multiple cases are found with the search criteria.
- NoneFound - No cases are found with the search criteria.

Restrictions

None. This can be used in any type of flow and any channel.

Corresponding block in the UI

Flow block: Cases

InvokeFlowModule

Invokes a flow module. *Flow modules* are reusable sections of a flow. You use them to extract repeatable logic across your flows, and create common functions. For more information about flow modules, see Flow modules for reusable functions, in the *Connect Customer Administrator Guide*.

Parameter object

```
{
    "FlowModuleId": The flow module ID or flow module ARN to be invoked. May be defined
 statically or dynamically.
}
```

Results and conditions

None.

Errors

`NoMatchingError` if no other Error matches.

Restrictions

This action is supported by all channels and only supports Inbound flow types.

Corresponding block in the UI

Flow block: Invoke module.

StartOutboundChatContact

Initiate an outbound chat contact to a customer. Only SMS chats are supported. For more information, see the StartOutboundChatContact in the *Connect Customer API Reference*.

Parameter object

```
{
    "SourceEndpoint": {
        "Address": ConnectPhoneNumberArn of the SourceEndpoint,
        "Type": Type of the SourceEndpoint, currently only supports
 CONNECT_PHONENUMBER_ARN
    },
    "DestinationEndpoint": {
        "Address": E164 phone number of the DestinationEndpoint,
        "Type": Type of the DestinationEndpoint, currently only supports
 TELEPHONE_NUMBER
    },
    "ContactFlowArn": ContactFlowArn of the flow to be executed for the new outbound
 chat,
    "ContactSubtype": Subtype of the new chat contact, currently only supports
 connect:SMS,
    "InitialSystemMessage": [Optional] {
        "Content": content of the initial system message to be sent to the
 DestinationEndpoint
    },
    "RelatedContact": [Optional] Only supported value is CURRENT
}
```

Results and conditions

None. No conditions are supported.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

- This action only supports `connect:SMS` as the **ContactSubtype** currently.
- This action only supports `CONNECT_PHONENUMBER_ARN` as the Type of **SourceEndpoint** currently.

- This action only supports `TELEPHONE_NUMBER` as the Type of **DestinationEndpoint** currently.

Corresponding block in the UI

Send message

TagContact

Sets a collection of tag to the current contact. With this type of operation, either all tags are set or none are set.

Each tag on a contact is a key-value pair, that is, it is composed of a key and a value. There are two types of tags: user-defined tags and system tags. You create a user-defined tag. A system tag is automatically created by AWS services. A system tag is prefixed with `aws:`. You cannot change it.

The TagContact action parameters correspond to user-defined tags that are applied on a contact.

Parameter object

```
{
"Tags": { an Object that holds the tags to be set.
       "Key1":"Value1" Both the key and value may be defined statically or dynamically.
        }
}
```

Results and conditions

None.

Errors

None.

Restrictions

None. This can be used in any type of flow and any channel.

Corresponding block in the UI

Contact tags

TransferContactToAgent

Ends the current flow and transfers the customer to an agent. If the agent is already with someone else, the contact is disconnected. Transfer contact to agent works only for voice interactions.

Parameter object

No parameters are expected.

Results and conditions

None.

Errors

None.

Restrictions

This action is supported in only transfer to agent and transfer to queue flows.

Corresponding block in the UI

Transfer to agent

TransferContactToQueue

This action places a contact that is not already in a queue into the contact's TargetQueue. If the contact has already been put into a queue (meaning that it is currently being routed to an agent, being joined to an agent, or is connected to an agent), the action fails.

Parameter object

No parameters are expected.

Results and conditions

None.

Errors

- QueueAtCapacity - if the destination queue is at capacity and the contact cannot be queued within it.

- NoMatchingError - if no other Error matches.

Restrictions

This action is supported in inbound contact flows and transfer flows. It is not supported in whisper flows, customer queue flows, or hold flows.

Corresponding block in the UI

Transfer to queue

UnTagContact

Removes a collection of tags on the current contact. With this type of operation, either all tags are set or none are set.

You cannot remove system-defined tags. You can only remove already existing user-defined tags from a contact.

Parameter object

```
{
"TagKeys": [ an Object that holds the tag-keys for the tags to be removed.
     "Key1" Key(s) can only be set statically.
      ]
}
```

Results and conditions

None.

Errors

- NoMatchingError - if no other Error matches

Restrictions

This action is supported across all the Connect Customer media channels.

This action can be used in flows of all types.

Corresponding block in the UI

Contact tags

UpdateCase

Updates an existing case by providing the case's id and the fields that should be updated.

Parameter object

```
{
  “LinkContactToCase” : "true" or "false", If set to true, cases open automatically
 when the agent accepts the contact.
  “CaseId”: the unique identifier of the case
  “CaseRequestFields”: An optional map of case fields to be updated. Keys should be
 fields from the Cases domain. Values can be static or dynamic.
}
```

Results and conditions

None.

Errors

- ContactNotLinked - If you specify to link the contact to case, then this error branch will appear. It might be that the contact was not linked after the case is retrieved (partial success/partial failure). If this happens, then the flow will follow this branch.
- NoMatchingError - An error was encountered while trying to find the case. This may be due to a system error, or how Update case is configured.

Restrictions

None. This can be used in any type of flow and any channel.

Corresponding block in the UI

Flow block: Cases

UpdateContactAttributes

Sets a collection of contact attributes on either the current contact or the related contact. With this type of operation, either all attributes are set or none are set.

Parameter object

```
{
    "Attributes": { an Object that holds the attributes to be set.
        "Key": "Value" Both the key and value may be defined statically or dynamically.
    },
    "TargetContact": either "Current" or "Related". Must be defined statically.
}
```

Results and conditions

None.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

None. This can be used in any type of flow and any channel.

Corresponding block in the UI

Set contact attributes

UpdateContactCallbackNumber

Updates the contact callback number, which is the number used by the CreateCallbackContact action. This value defaults to the customer participant caller ID if this action is never used.

Parameter object

```
{
    "CallbackNumber": The callback number to set. Must be a single, valid JSONPath
 reference, and cannot be set statically.
}
```

Results and conditions

None.

Errors

- InvalidCallbackNumber - The callback number specified was not a valid (e.164) phone number.
- CallbackNumberNotDialable - The callback number specified is not dialable by the instance.

Restrictions

This is supported only in contact flows, transfer flows, and customer queue flows. This is not supported in whispers or hold flows.

Corresponding block in the UI

Set callback number

UpdateContactData

Sets a collection of connect defined attributes on specified contact. With this type of operation, either all attributes are set or none are set.

Parameter object

```
"Parameters":
{
        "Name": [Optional] The name of the contact. It is a string. May be set
 statically or dynamically.
       "Description": [Optional] The description of the contact. It is a string.
       "LanguageCode": [Optional] The language to use for current contact.
       "CustomerId": [Optional] The customer id associated with the contact.
       "References": [Optional] { an Object that holds the references to be set.
         "Type": "Value" Both the key and value may be defined statically or
 dynamically.
       },
       "IsVoiceIdStreamingEnabled": [Optional] Enable to start streaming audio from
 customer channel to Voice ID. It is a string. "TRUE" and "FALSE" are the only valid
 values.
       "IsVoiceAuthenticationEnabled": [Optional] Enable authentication by comparing
 voiceprint of the caller to the enrolled voiceprint of the claimed identity.It is a
 string. "TRUE" and "FALSE" are the only valid values.
       "IsFraudDetectionEnabled": [Optional] Enable detection for impersonation
 attempts and presence of known fraudsters. It is a string. "TRUE" and "FALSE" are the
 only valid values.
```

```
        "VoiceAuthenticationThreshold": [Optional] Threshold to validate against
  confidence score of a voice match. It is a string. Value must be between 0 and 100.
        "VoiceAuthenticationResponseTime": [Optional] Define required minimum caller
  speech seconds for voice authentication. It is a string. Value must be between 5 and
  10.
        "FraudDetectionThreshold": [Optional] The threshold you set for fraud detection
  is used to measure risk. Scores higher than the threshold are reported as higher risk.
  Scores lower than the threshold are reported as lower risk. Raising the threshold
  lowers false positive rates (makes result more certain), but raises false negative
  rates. It is a string. Value must be between 0 and 100.
        "WatchlistId": [Optional] Identifier of watchlist to use when evaluating the
  voice session. It is a string. Value must be between 0 and 100.
        "WisdomSessionArn": [Optional] A session ARN provided by Connect Customer
  Wisdom for agent assistance. It is a string.
        "TargetContact": [Required] Target contact on which given attributes should be
  set. It is a string. "Current" or "Related" are the only valid values.
}
```

Results and conditions

None. No conditions are supported.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

This action is supported on all channels and in all flow types.

Corresponding blocks in the UI

- Set contact attributes

UpdateContactEventHooks

Sets one or more contact event hooks, which are flows associated with contact events, such as customer whisper or agent hold. For more information, see Contact records data model. The following event hooks are valid:

- AgentHold

- AgentWhisper
- CustomerHold
- CustomerQueue
- CustomerRemaining
- CustomerWhisper
- DefaultAgentUI
- DisconnectAgentUI
- PauseContact
- ResumeContact

Parameter object

```
{
    "EventHooks": { an Object that holds the event hooks to be set. Only one entry may
 be present in this map.
        "Key": "Value" - the event hook to be set where the key is the event type and
 the value is the flow ID or ARN to run when that event occurs. Keys must be defined
 statically.
    }
}
```

Results and conditions

None.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

This is supported in all types of flows.

Corresponding blocks in the UI

- Set customer queue flow
- Set event flow
- Set hold flow

- Set whisper flow

UpdateContactMediaProcessing

Allows customers to configure their own Lambda processor, which will be applied to in-flight messages.

Parameter object

```
{
    "ChatProcessor": { // Configuration for bring-your-own-processor for chat channel.
        "ProcessingEnabled": either "True" or "False". Must be set statically.
 Determines whether to enable custom Lambda processing for chat messages.
        "LambdaProcessorARN": The ARN of the Lambda function to process chat messages.
 Must be set statically. Format: arn:aws:lambda:region:account-id:function:function-
name
        "ChatProcessorSettings": { // An object that holds chat processor behavior
 settings
            "DeliverUnprocessedMessages": either "True" or "False". Must be set
 statically. Determines whether to deliver messages that fail Lambda processing.
        }
    }
}
```

Results and conditions

None.

Errors

- **NoMatchingError** - if no other Error matches. Must always be defined.
- **ChannelMismatch** - if the media channel that initiated the contact is not the same as the one defined in the action. As of now, only chat is supported in this action.

Corresponding block in the UI

Set recording, analytics and processing behavior

UpdateContactMediaStreamingBehavior

Enables or disables contact media streaming for a set of participants.

Parameter object

```
{
    "MediaStreamingState": One of "Enabled" or "Disabled". Must be specified
 statically.
    "Participants": [  A list of participants to include in the stream if enabling the
 stream, or disable if disabling the stream
        {
            "ParticipantType": The type of participant to stream. Currently, only
 "Customer" is supported. Must be defined statically.
            "MediaDirections": [ ] A list of the directions of media to include in the
 stream - "From" and "To".  Must be defined statically.
        }
    ],
    "MediaStreamType": The type of media to enable or disable from the stream.
 Currently, only "Audio" is supported. Must be defined statically.
}
```

Results and conditions

None.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

This is supported in contact flows, customer queue flows, transfer flows, and whisper flows. It is not supported in hold flows.

This is supported only by the voice channel.

Corresponding block in the UI

Start media streaming and Stop media streaming

UpdateContactRecordingAndAnalyticsBehavior

Sets contact recording behavior, including analysis behavior and which participants of the contact to record.

Parameter object

```
{
    // Only ONE of the following channel behavior objects can be defined per
 configuration

   "ChatBehavior": { // Configuration for chat channel interactions
       "ChatAnalyticsBehavior": {
           "Enabled": either "True" or "False". Must be set statically.
           "AnalyticsLanguage": Must be one of languages supported by Contact Lens
 analysis. Can be set dynamically. Use the format xx-XX, for example, en-US for US
 English.
           "ConversationalAnalyticsRedactionConfiguration": {
               "Enabled": either "True" or "False". Must be set statically. Determines
 whether to redact sensitive data in Contact Lens output.
               "RedactionResults": either "RedactedAndOriginal" or "RedactedOnly".
 Can be set dynamically. Determines whether to provide both versions or only redacted
 output.
               "RedactionMaskMode": either "EntityType" or "PII". Must be set
 statically.
               "RedactionEntities": [ ] a list of entity types to redact from the
 analytics output. Must be set statically.
           },
           "InFlightChatRedactionConfiguration": { // In-flight redaction for chat
 messages as they are sent. Only available for chat channel.
               "Enabled": either "True" or "False". Must be set statically. Determines
 whether to redact sensitive data in real-time chat messages.
               "RedactionMaskMode": either "EntityType" or "PII". Must be set
 statically.
               "RedactionEntities": [ ] a list of entity types to redact from chat
 messages in-flight. Must be set statically.
               "DeliverUnprocessedMessages": either "True" or "False". Must be set
 statically. Determines whether to deliver messages that fail redaction.
           },
           "AnalyticsModes": ["ContactLens"]. Can only be set to ContactLens for chat
 channel.
           "SentimentConfiguration": { // Optional parameter to enable sentiment
 analysis
               "Enabled": either "True" or "False". Must be set statically.
           },
           "SummaryConfiguration": { // Optional parameter to enable post-contact
 summary
               "SummaryModes": ["PostContact", "AutomatedInteraction"]. Can be set to
 "PostContact" or "AutomatedInteraction", or both for chat channel.
```

```
            }
        }
    },

    "VoiceBehavior": { // Configuration for voice channel interactions
        "VoiceRecordingBehavior": {
            "RecordedParticipants": [ ] a list of participants to record, chosen from
"Agent" and "Customer". An empty list disables recording. Must be set statically.
            "IVRRecordingBehavior": Can be either "Enabled" or "Disabled". Must be set
statically.
        },
        "VoiceAnalyticsBehavior": { // An object that holds voice analytics settings.
Can only be set if RecordedParticipants contains both Agent and Customer.
            "Enabled": either "True" or "False". Must be set statically.
            "AnalyticsLanguage": Must be one of languages supported by Contact Lens
analysis. Must be set statically. Use the format xx-XX, for example, en-US for US
English.
            "ConversationalAnalyticsRedactionConfiguration": { // Redaction settings
for analytics output (transcripts, audio files)
                "Enabled": either "True" or "False". Must be set statically. Determines
whether to redact sensitive data in Contact Lens output.
                "RedactionResults": either "RedactedAndOriginal" or "RedactedOnly".
Can be set dynamically. Determines whether to provide both versions or only redacted
output.
                "RedactionMaskMode": either "EntityType" or "PII". Must be set
statically.
                "RedactionEntities": [ ] a list of entity types to redact from the
analytics output. Must be set statically.
            },
            "AnalyticsModes": ["RealTime", "AutomatedInteraction"]. Can be set to one
of "RealTime" or "PostContact" and also can include "AutomatedInteraction" for voice
channel.
            "SentimentConfiguration": { // Optional parameter to enable sentiment
analysis
                "Enabled": either "True" or "False". Must be set statically.
            },
            "SummaryConfiguration": { // Optional parameter to enable post-contact
summary
                "SummaryModes": ["PostContact", "AutomatedInteraction"]. Can be set
to "PostContact" or "AutomatedInteraction", or both for voice channel. Must be set
statically.
            }
        }
    },
```

```
    "ScreenRecordingBehavior": { // Optional configuration for agent screen recording.
 Can be defined independently or alongside any channel behavior.
        "ScreenRecordedParticipants": [ ] a list of participants for which to record
 their screen, can only include "Agent". An empty list disables screen recording. Must
 be set statically.
    }
}
```

Notes

- `AnalyticsRedactionMaskMode`: Optional, String. Allowed values:
  - `PII`: All PII data is replaced with `[PII]`. For example, Jane Doe is replaced with `[PII]`
  - `EntityType`: Each PII entity is replaced with its type. For example, Jane Doe is replaced with `[NAME]`.
  - If no value is provided, the default `PII` is used.
- `AnalyticsRedactionEntities`: Optional, Array of strings.
  - Valid values include: "BANK_ACCOUNT_NUMBER", "BANK_ROUTING", "CREDIT_DEBIT_NUMBER", "CREDIT_DEBIT_CVV", "CREDIT_DEBIT_EXPIRY", "INTERNATIONAL_BANK_ACCOUNT_NUMBER", "PIN", "SWIFT_CODE", "CA_HEALTH_NUMBER", "UK_NATIONAL_HEALTH_SERVICE_NUMBER", "CA_SOCIAL_INSURANCE_NUMBER", "SSN", "UK_NATIONAL_INSURANCE_NUMBER", "PASSPORT_NUMBER", "DRIVER_ID", "IN_AADHAAR", "NAME", "AGE", "EMAIL", "PHONE", "ADDRESS", "US_INDIVIDUAL_TAX_IDENTIFICATION_NUMBER", "UK_UNIQUE_TAXPAYER_REFERENCE_NUMBER", "IN_PERMANENT_ACCOUNT_NUMBER", "IN_NREGA", "AWS_ACCESS_KEY", "AWS_SECRET_KEY", "IP_ADDRESS", "MAC_ADDRESS", "PASSWORD", "URL", "USERNAME", "LICENSE_PLATE", "VEHICLE_IDENTIFICATION_NUMBER", "IN_VOTER_NUMBER", "DATE_TIME", "AGENT_DISPLAY_NAME", "CUSTOMER_DISPLAY_NAME", "ATTACHMENT_NAME".
  - An empty array is not allowed.
  - If `AnalyticsRedactionEntities` is not present, the default "redact all PII data" is used.
  - The following redaction entities are not supported for chat in-flight redaction:
    - IN_PERMANENT_ACCOUNT_NUMBER
    - IN_NREGA
    - IN_VOTER_NUMBER
    - IN_AADHAAR

- DATE_TIME
- CUSTOMER_DISPLAY_NAME
- AGENT_DISPLAY_NAME
- ATTACHMENT_NAME

For more information on sensitive data redaction, see Enable redaction of sensitive data in the *Connect Customer Administrator's Guide*.

For more information on in-flight chat redaction, see Enable in-flight sensitive data redaction and message processing.

For a list of languages supported by Contact Lens post-call analysis, see Contact Lens supported languages. For the 4-character language code to use, see Supported languages in the *Amazon Transcribe Developer Guide*.

Results and conditions

None.

Errors

- **NoMatchingError** - if no other Error matches. Must always be defined.
- **ChannelMismatch** - if the media channel that initiated the contact is not the same as the one defined in the action. For screen recording, any channel other than voice, chat or tasks would result in this branch being taken. Must always be defined.
- **InFlightRedactionConfigurationFailed** - if starting/stopping in-flight chat redaction fails. Must be defined if chat behavior is defined in action.

Corresponding block in the UI

Set recording, analytics and processing behavior

UpdateContactRecordingBehavior

Sets contact recording behavior, including analysis behavior and which participants of the contact to record.

Parameter object

```
{
    "RecordingBehavior": { an object that holds the recording behavior
        "RecordedParticipants": [ ] a list of participants to record, chosen from
 "Agent" and "Customer". An empty list disables recording. Must be set statically.
        "ScreenRecordedParticipants": [ ] a list of participants for which to record
 their screen, can only include "Agent". Must be set statically.
        "IVRRecordingBehavior": Can be either "Enabled" or "Disabled". Must be set
 statically.
    },
    "AnalyticsBehavior": { an object that holds the analytics behavior. Can only be set
 if the RecordedParticipants contains both Agent and Customer
        "Enabled": either "True" or "False". Must be set statically.
        "AnalyticsLanguage": Must be one of languages supported by Contact Lens post-
call analysis. Must be set statically. Use the format xx-XX, for example, en-US for US
 English.
        "AnalyticsRedactionBehavior": either Enabled or Disabled. Defaults to Disabled
 if not set. Determines whether to redact sensitive data, such as personal information,
 in the Contact Lens output file and audio recording.
        "AnalyticsRedactionResults": either "RedactedAndOriginal" or "RedactedOnly".
 Can be set dynamically. Determines whether the customer gets both the redacted and
 the original transcripts and audio files, or just the redacted transcripts and audio
 files.
        "AnalyticsRedactionMaskMode": either "EntityType" or "PII".  Must be set
 statically.
        "AnalyticsRedactionEntities": ["BANK_ACCOUNT_NUMBER", "BANK_ROUTING",
 "CREDIT_DEBIT_NUMBER", "CREDIT_DEBIT_CVV", "CREDIT_DEBIT_EXPIRY",
 "INTERNATIONAL_BANK_ACCOUNT_NUMBER", "PIN", "SWIFT_CODE", "CA_HEALTH_NUMBER",
 "UK_NATIONAL_HEALTH_SERVICE_NUMBER", "CA_SOCIAL_INSURANCE_NUMBER", "SSN",
 "UK_NATIONAL_INSURANCE_NUMBER", "PASSPORT_NUMBER", "DRIVER_ID", "IN_AADHAAR", "NAME",
 "AGE", "EMAIL", "PHONE", "ADDRESS", "US_INDIVIDUAL_TAX_IDENTIFICATION_NUMBER",
 "UK_UNIQUE_TAXPAYER_REFERENCE_NUMBER", "IN_PERMANENT_ACCOUNT_NUMBER", "IN_NREGA",
 "AWS_ACCESS_KEY", "AWS_SECRET_KEY", "IP_ADDRESS", "MAC_ADDRESS", "PASSWORD", "URL",
 "USERNAME", "LICENSE_PLATE", "VEHICLE_IDENTIFICATION_NUMBER", "IN_VOTER_NUMBER",
 "DATE_TIME", "AGENT_DISPLAY_NAME", "CUSTOMER_DISPLAY_NAME", "ATTACHMENT_NAME"],
        "ChannelConfiguration": {
          "Chat": {
            "AnalyticsModes": ["ContactLens"]  Can only be set to ContactLens
          },
         "Voice": {
            "AnalyticsModes": ["RealTime"]  Can be set to RealTime and PostContact
          }
       },
```

```
        "SummaryConfiguration": {
            "SummaryModes": ["PostContact"] Optional parameter to enable post-contact
 summary. At present we only support "PostContact".
        },
         "SentimentConfiguration": {
            "Enabled": either "True" or "False". Must be set statically.
        }
    }
}
```

Notes

- `AnalyticsRedactionMaskMode`: Optional, String. Allowed values:
  - `PII`: All PII data is replaced with `[PII]`. For example, Jane Doe is replaced with `[PII]`
  - `EntityType`: Each PII entity is replaced with its type. For example, Jane Doe is replaced with `[NAME]`.
  - If no value is provided, the default `PII` is used.
- `AnalyticsRedactionEntities`: Optional, Array of strings.
  - Valid values include: "BANK_ACCOUNT_NUMBER", "BANK_ROUTING", "CREDIT_DEBIT_NUMBER", "CREDIT_DEBIT_CVV", "CREDIT_DEBIT_EXPIRY", "INTERNATIONAL_BANK_ACCOUNT_NUMBER", "PIN", "SWIFT_CODE", "CA_HEALTH_NUMBER", "UK_NATIONAL_HEALTH_SERVICE_NUMBER", "CA_SOCIAL_INSURANCE_NUMBER", "SSN", "UK_NATIONAL_INSURANCE_NUMBER", "PASSPORT_NUMBER", "DRIVER_ID", "IN_AADHAAR", "NAME", "AGE", "EMAIL", "PHONE", "ADDRESS", "US_INDIVIDUAL_TAX_IDENTIFICATION_NUMBER", "UK_UNIQUE_TAXPAYER_REFERENCE_NUMBER", "IN_PERMANENT_ACCOUNT_NUMBER", "IN_NREGA", "AWS_ACCESS_KEY", "AWS_SECRET_KEY", "IP_ADDRESS", "MAC_ADDRESS", "PASSWORD", "URL", "USERNAME", "LICENSE_PLATE", "VEHICLE_IDENTIFICATION_NUMBER", "IN_VOTER_NUMBER", "DATE_TIME", "AGENT_DISPLAY_NAME", "CUSTOMER_DISPLAY_NAME", "ATTACHMENT_NAME".
  - An empty array is not allowed.
  - If `AnalyticsRedactionEntities` is not present, the default "redact all PII data" is used.

For more information on sensitive data redaction, see Enable redaction of sensitive data in the *Connect Customer Administrator's Guide*.

For a list of languages supported by Contact Lens post-call analysis, see Contact Lens supported languages. For the 4-character language code to use, see Supported languages in the *Amazon Transcribe Developer Guide*.

Results and conditions

None.

Errors

None.

Restrictions

This is supported only in contact flows, transfer flows, outbound whispers, and customer queue flows. This is not supported in agent/customer whispers or hold flows.

Analytics is only supported by the voice channel.

Corresponding block in the UI

Set recording and analytics behavior

ResumeContact

Resumes a contact from a paused state.

Parameter object

No parameters are expected.

Results and conditions

None.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

None. This action can be used everywhere.

UpdateContactRoutingBehavior

Updates the contact's routing details. This can move the contact forward or backward in queue, or specify a queue priority.

Parameter object

```
{
  "QueuePriority": An integer that represents the queue priority to be applied to
 the contact (lower priorities are routed preferentially). Cannot be specified if
 QueueTimeAdjustmentSeconds is specified. Must be statically defined, must be larger
 than zero, and a valid integer value.
  "QueueTimeAdjustmentSeconds": An integer that represents the queue time adjust
 to be applied to the contact, in seconds (longer / larger queue times are routed
 preferentially). Cannot be specified if QueuePriority is specified. Must be statically
 defined and a valid integer value.
}
```

Results and conditions

None.

Errors

None.

Restrictions

This is supported only in inbound contact flows. It is not supported in transfer flows, whisper flows, customer queue flows, or hold flows.

Corresponding block in the UI

Change routing priority / age

UpdateContactTargetQueue

Sets the contact's TargetQueue. This is the queue is used by all other instructions that check a queue implicitly, and for TransferContactToQueue.

Parameter object

```
{
```

```
  "QueueId": [Optional] A queue ID or queue ARN. If AgentId is specified, this may
 not be specified. This must be either defined fully statically or as a single valid
 JSONPath identifier.
  "AgentId": [Optional] An agent ID or agent ARN, representing an agent queue. If
 QueueId is specified, this may not be specified. This must be either defined fully
 statically or as a single valid JSONPath identifier.
}
```

Results and conditions

None.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

This action is supported only in inbound contact flows and transfer flows. It is not supported in whisper flows, hold flows, or customer queue flows.

Corresponding block in the UI

Set working queue

UpdateContactTextToSpeechVoice

Updates the Amazon Polly voice used by text-to-speech for voice contacts (message with text-to-speech, or Amazon Lex bots). This defaults to Joanna if this action is never run.

Parameter object

```
{
  "TextToSpeechVoice": A string holding the name of an Amazon Polly voice. May be
 defined statically or dynamically. If this is an invalid text to speech voice, text to
 speech is no longer function for this contact.
  "TextToSpeechEngine": The engine associated with the Amazon Polly voice. May be
 defined statically or dynamically.
  "TextToSpeechStyle" : The speech style associated with the Amazon Polly Voice. It
 could be None, Coversational, or Newscaster. May be defined statically or dynamically.
```

```
}
```

Results and conditions

Results in error if voice or engine are invalid, or if the selected voice does not support the selected engine.

Errors

- NoMatchingError - if no other Error matches. Must always be defined.

Restrictions

None. This action is supported in all flow types, and across all channels.

Corresponding block in the UI

Set voice

UpdatePreviousContactParticipantState

This action is primarily used to prevent previous participants on the contact from observing the contact. Common use cases are disconnecting the agent that initiates a transfer when they transfer a contact to a secure destination, or putting the agent on hold when transferring to a quick connect that securely gathers customer input such as credit card numbers.

Parameter object

```
{
    "PreviousContactParticipantState": One of ["AgentOnHold", "CustomerOnHold",
 "OffHold"], which are only supported for voice contacts.
}
```

Execution results and conditions

None.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

This action is supported only in inbound contact flows and transfer flows.

Corresponding block in the UI

Hold customer or agent

Flow control actions in the Connect Customer Flow language

These actions don't have any side effects and are only used to determine the path through a flow. Certain data may not be available (such as contact data, if the action is determining its path based on contact data). These actions generally work in every circumstance.

A flow control action is an action that:

- Does not need a contact or a participant to succeed.
- Controls the behavior of the flow, by either enabling or disabling flow behavior (such as logging) or by choosing a branch when the flow runs.

Contents

CheckHoursOfOperation

Returns whether the specified hours of operation object (or the hours of operation object associated with the current queue if no hours of operation is referenced) is in hours or out of hours as its result, allowing comparisons against it.

Parameter object

```
{
  "HoursOfOperationId": [Optional] An hours of operation ID or hours of operation ARN.
 *Must be either fully static or fully dynamic*. If not specified, the TargetQueue's
 hours of operation for the contact are used
}
```

Results and conditions

True or **False** based on whether the hours of operation object specified is in hours or out of hours. There must be a Condition provided for Equals **True** and a Condition for Equals **False**, and no other conditions.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

This action is available in inbound flows, transfer flows, and customer queue flows. It is not available to hold flows or to whisper flows.

Corresponding block in the UI

Check hours of operation

CheckMetricData

A shortcut single action to avoid using GetMetricData and Compare for a set of simple metrics. This action loads the specified metric data for the specified queue, and allows comparisons to the loaded value. For example, it loads number of contacts in queue, age of oldest contact in queue, number of agents staffed on the queue, number of agents available on the queue, or number of agents online on the queue.

Parameter object

```
{
  "MetricType": One of [NumberOfAgentsAvailable, NumberOfAgentsStaffed,
 NumberOfAgentsOnline, OldestContactInQueueAgeSeconds, NumberOfContactsInQueue].
 **Dynamic values are not supported**,
  "QueueId": [Optional] A queue ID or queue ARN. If AgentId is specified, this may not
 be specified. *Dynamic values are supported*,
  "AgentId": [Optional] An agent ID or agent ARN, representing an agent queue. If
 QueueId is specified, this may not be specified. *Dynamic values are supported*. If
 neither this nor QueueId are specified, the contact TargetQueue is used
}
```

Execution results and conditions

A number, representing the value of the metric that was requested. This can be used for conditions. If the MetricType is NumberOfAgents* then the only supported condition is "NumberGreaterThan 0", otherwise Equals and any Number* Operands are allowed.

Errors

- NoMatchingError - if no other Error matches.
- NoMatchingCondition - if no other Condition matches (only supported if the MetricType is OldestContactInQueueAgeSeconds or NumberOfContactsInQueue).

Restrictions

This action is only usable in flows, queue and agent transfers, and customer queue flows. It is not available in any type of whisper or hold flows.

Corresponding block in the UI

- Check staffing
- Check queue status

CheckOutboundCallStatus

Engages with the output provided by an answering machine, and provides branches to route the contact accordingly.

Parameter object

```
{

}
```

Execution results and conditions

- "CallAnswered" if the call has been answered by a person.
- "VoicemailBeep" if Connect Customer identifies that the call ended in a voice mail and it detects a beep.
- "VoicemailNoBeep" if Connect Customer identifies that call ended in a voicemail, but it doesn't detect a beep, or the beep is unknown.
- "NotDetected" if Connect Customer could not detect whether there is a voicemail. This happens when Connect Customer is unable to make a positive determination of whether a call was answered by a live voice or an answering machine. Typical situations that result in this state include long silences or excessive background noise.

Conditions are supported, but only the "Equals" operator is supported. "CallAnswered", "VoicemailBeep" , "VoicemailNoBeep" and "NotDetected" are the only supported operands.

Errors

- NoMatchingError if no condition matches.

Restrictions

This action works with Connect Customer outbound campaigns only.

Corresponding block in the UI

Check call progress

CheckVoiceId

Checks the enrollment status, voice authentication or fraud detection results of the voice analysis returned by Voice ID.

Parameter object

```
{
 "CheckVoiceIdOption": "enrollmentStatus"
}
```

```
{
 "CheckVoiceIdOption": "voiceAuthentication"
}
```

```
{
 "CheckVoiceIdOption": "fraudDetection"
}
```

Execution results and conditions

The checkVoiceId action returns results of the voice analysis and the status returned by Voice ID. The following is returned when CheckVoiceIdOption input is set as:

- **enrollmentStatus**:
  - **Enrolled**: The caller is enrolled in voice authentication.
  - **Not enrolled**: The caller has not yet been enrolled in voice authentication. When this status is returned, for example, you may want to directly route the call to an agent for enrollment.
  - **Opted out**: The caller has opted out of voice authentication.

  You are not charged for checking enrollment status.
- **voiceAuthentication**:
  - **Authenticated**: The caller's identity has been verified. That is, the authentication score is greater than or equal to the threshold (default threshold of 90 or your custom threshold).
  - **Not authenticated**: The authentication score is lower than threshold that you configured.
  - **Inconclusive**: Unable to analyze a caller's speech for authentication. This is usually because Voice ID did not get the required 10 seconds to provide a result for authentication.
  - **Not enrolled**: The caller has not yet been enrolled in voice authentication. When this status is returned, for example, you may want to directly route the call to an agent for enrollment.
  - **Opted out**: The caller has opted out of voice authentication.

  You are not charged if the result is **Inconclusive**, **Not enrolled** or **Opted out**.

- **fraudDetection**:
  - **High risk**: The risk score meets or exceeds the set threshold.
  - **Low risk**: The risk score did not meet the set threshold.
  - **Inconclusive**: Unable to analyze a caller's voice for detection of fraudsters in a watchlist.

Errors

- NoMatchingError if no condition matches.

Restrictions

Only supported for voice channel. If used with the chat or task channels, the action takes the **Error** branch.

Corresponding block in the UI

Check Voice ID

Compare

Allows comparisons against the specified value.

Parameter object

```
{
  "ComparisonValue": Any **single** JSONPath identifier that is valid for the flow data
 object
}
```

Execution results and conditions

The value specified for comparison. This can be used for conditions.

Errors

- NoMatchingCondition - if no other Condition matches.

Restrictions

This action is available in every type of flow.

Corresponding block in the UI

Check contact attributes

DistributeByPercentage

Returns a random number between 1 and 100 (inclusive) as its result, allowing comparisons against it.

Parameter object

```
{

}
```

Results and conditions

A number between 1 and 100, inclusive, chosen randomly. Comparisons are supported, but they must be a chain of NumericLessThan comparisons, with each subsequent comparison checking the previous value, plus the percentage that is desired to go down this next action, and no Comparison comparing a value larger than 100.

Errors

- NoMatchingCondition if no Condition matches. This is the default option in the flow editor.

Restrictions

This action is available in inbound flows, transfer flows, and customer queue flows. It is not available to hold flows or to whisper flows.

Corresponding block in the UI

Distribute by percentage

EndFlowExecution

Finishes flow, but does not explicitly disconnect the participant. The participant may be disconnected by contact logic after this. For example, if a flow ends before the contact is put into queue, ending the flow results in the contact being ended.

Parameter object

```
{

}
```

Results and conditions

None. No conditions are supported.

Errors

None. This is always a terminal action.

Restrictions

This action is available only in whisper flows and customer queue flows. It is not available in flows, hold flows, or transfer flows.

Corresponding block in the UI

End flow / Resume

GetMetricData

Loads real time queue metrics for the queue specified by queue ID, agent ID (for agent queues), or the target queue, and makes them available on the flow run data. May be extended in the future to allow getting historical metric data in addition to current metric data, and to getting agent metrics in addition to queue metrics.

Parameter object

```
{
  "QueueId": [Optional] A queue ID or queue ARN. If AgentId is specified, this may not
 be specified. *Dynamic values are supported*,
  "AgentId": [Optional] An agent ID or agent ARN, representing an agent queue. If
 QueueId is specified, this may not be specified. *Dynamic values are supported*
  "QueueChannel": [Optional] Either "Voice" or "Chat". Can be set dynamically.
 Determines the channel for which metrics are returned. If not specified, metrics are
 returned for all channels.
}
```

Execution results and conditions

None. No conditions are supported.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

This action is available in every type of flow.

Corresponding block in the UI

Get queue metrics

Loop

When the same action (the same Action Identifier) is run multiple times, this block returns a result of "NotDone" a number of times equal to the specified loop count, then "Done" once, then reset.

Parameter object

```
{
    "LoopCount": Number of times to loop, must be between 0 and 100 (inclusive). Must
 either be fully static or fully dynamic.
}
```

Execution results and conditions

"ContinueLooping" if the loop should continue. "DoneLooping" if the loop should finish. Conditions are supported, there must be a Condition provided for Equals ContinueLooping and for Equals DoneLooping, and no other Conditions can be specified.

Errors

None.

Restrictions

This is supported in every type of flow.

Corresponding block in the UI

Loop

StartVoiceIdStream

Sends audio to Connect Customer Voice ID to verify the caller's identity and match against fraudsters in watchlist, as soon as the call is connected to a flow.

Parameter object

```
{

}
```

Execution results and conditions

None. No conditions are supported.

Errors

- NoMatchingError if no condition matches.

Restrictions

Only supported for the voice channel. If used with the chat or task channels, the action takes the **Error** branch. Not supported in hold flows.

Corresponding block in the UI

Set Voice ID

TransferToFlow

Execution jumps to a different flow, and continues running at that flow's beginning.

Parameter object

```
{
    "ContactFlowId": A flow ID or flow ARN. *Must be either fully static or a single
 valid JSONPath identifier*
```

```
}
```

Execution results and conditions

None.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

This action is available in inbound flows and transfer flows. It is not available to hold flows, customer queue flows, or whisper flows.

Corresponding block in the UI

Transfer to flow

UpdateFlowAttributes

Sets a collection of attributes on the current flow. These attributes are not carried over to the subsequent flows. With this type of operation, either all attributes are set or none are set.

Parameter object

```
{
    ""FlowAttributes": { An Object that holds the attributes to be set. Keys are of
 type String, Values are of type FlowAttribute
        "Type": {
            FlowAttribute" : "Value"
        }
    }
}
```

Results and conditions

None. No conditions are supported.

Errors

None.

Restrictions

This action is supported on all channels and in all flow types.

Corresponding block in the UI

Set contact attributes

UpdateFlowLoggingBehavior

Enables or disables flow logging. If this is a flow, this same behavior remains unless it is overridden for the rest of the contact segment. It is also automatically inherited by new segments in the chain.

Parameter object

```
{
  "FlowLoggingBehavior": One of [Enabled,Disabled]. *Dynamic values are not supported*
}
```

Results and conditions

None. No conditions are supported.

Errors

None.

Restrictions

This action is available in every type of flow.

Corresponding block in the UI

Set logging behavior

UpdateRoutingCriteria

Sets the routing criteria for the contact.

Parameter object

```
{
    "RoutingCriteria": { Required. RoutingCriteria is a JSON object.
            "Steps": [{  Required. List of routing steps. When Connect Customer not
 find an available agent meeting the requirements in a step for a given step duration,
```

```
 the routing criteria will move on to the next step sequentially until a join is
 completed with an agent. When all steps are exhausted, the contact will be offered to
 any agent in the queue.  May be set statically or dynamically
                "Expression": { Required. An object to specify the expression of a
 routing step. It is a tagged union to specify expression for a routing step.
                    "AttributeCondition": {An object to specify the predefined
 attribute condition.
                        "Name": The name of predefined attribute. It is a string and
 has length constraints between 1-64.
                        "Value": The value of predefined attribute. It is a string and
 has length constraints between 1-64.
                        "ProficiencyLevel": The proficiency level of the condition. It
 is a float value. Valid values are: 1.0, 2.0, 3.0, 4.0 and 5.0
                        "ComparisonOperator": The operator of the condition. It is a
 string. Valid values: NumberGreaterOrEqualTo
                    }
                   "AndExpression": [List of routing expressions (attribute conditions)
 which will be AND-ed together.]
                },
                "Expiry": { An object to specify the expiration of a routing step.
                    "DurationInSeconds": The number of seconds to wait before expiring
 the routing step. Can be set only statically
                }
            }]
        }
}
```

Results and conditions

None. No conditions are supported.

Errors

NoMatchingError - if no other Error matches.

Restrictions

This action is supported on all channels and in only in Inbound flow, Customer Queue flow, Transfer to Agent flow, and Transfer to Queue flow types.

Corresponding block in the UI

Set routing criteria

Wait

Pauses the flow for a specified duration, or until a specified event happens, whichever happens first.

Parameter object

```
{
    "TimeoutSeconds": The amount of time to wait before the action finishes with the
 "WaitCompleted" result. This can be either statically defined, or a single valid
 JSONPath identifier. If defined statically, this must be a positive integer value no
 greater than 604800 (seven days),
    "Events": An optional list of all events that can trigger an interrupt. The
 supported events currently are "CustomerReturned" and "BotParticipantDisconnected".
 This must be defined statically.

}
```

Execution results and conditions

If an event interrupts the wait, the run result is the event that interrupted. If no event interrupts the Wait and the time elapses, the run result is WaitCompleted. Conditions are supported, but only the "Equals" operator is supported. "WaitCompleted" is always required operand, and every specified event is also required to be present as a condition operand.

Errors

- NoMatchingError - If no other Error matches.
- ParticipantNotFound - The supported event currently is "BotParticipantDisconnected".

Restrictions

This is supported in every type of flow, but is supported only by the chat channel.

Corresponding block in the UI

Wait

Interactions in the Connect Customer Flow language

Interactions actions have side effects, but they don't require a contact or a participant. They include actions such as invoking an AWS Lambda function. They generally work in every circumstance.

Contents

AssociateContactToCustomerProfile

Associate a contact to a customer profile. Customer Profiles must be enabled for your Connect Customer instance.

See AddProfileKey in the *Connect Customer Customer Profiles API Reference*.

Parameter object

A `ProfileId` and `ContactId` must be present.

```
{
    "ProfileRequestData": {
        "ProfileId": Profile being associated
        "ContactId": ContactId being associated
    },
   "ProfileResponseData": {
       None.
   }
}
```

Results and conditions

None. Conditions are not supported.

Errors

- NoMatchingError - if no other Error matches.

Corresponding block in the UI

Customer profiles block

CreateCallbackContact

Creates a new callback contact. If no customer number is specified, and this is run in context of a contact, the contact's CustomerCallbackNumber is used as the customer number. If you specify a ContactFlowId, then InitialCallDelaySeconds parameter is ignored.

Parameter object

```
{
    "QueueId": [Optional] A queue ID or queue ARN. The callback contact is routed with
 this queue, or if this is not specified, the contact's current TargetQueue. Must be
 specified fully statically or with a single valid JSONPath identifier.
    "AgentId": [Optional] An agent ID or agent ARN, representing an agent queue. If
 QueueId is specified, this may not be specified. This must be either defined fully
 statically or as a single valid JSONPath identifier.
    "InitialCallDelaySeconds": The amount of time, in seconds, to wait at a minimum
 before routing the callback contact. This gives the customer enough time to end their
 existing contact before being called back. Must be larger than 0, no greater than
 259,200 (three days), and an integer. Must be defined statically.
    "MaximumConnectionAttempts": The number of attempts at a maximum to connect this
 contact to a customer, if the callback is not answered. Must be larger than zero, and
 an integer. Must be defined statically.
    "RetryDelaySeconds": The minimum amount of time to wait, in seconds, between
 an unanswered callback attempt is made and the next attempt to reach the customer.
 Must be larger than 0, no greater than 259,200 (three days), and an integer. Must be
 defined statically.
    "ContactFlowId": [Optional] A contactFlow ID or contactFlow ARN. Callback contact
 created will execute this flow post creation, if specified. This must be either
 defined fully statically or as a single valid JSONPath identifier.
    "CallerId": [Optional] A caller ID representing the phone number to use for
 the callback. This is what the customer sees when dialed. Must be a valid phone
 number claimed in your Connect Customer instance. This must be either defined fully
 statically or as a single valid JSONPath identifier.
}
```

Results and conditions

None. No conditions are supported.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

This action is supported in contact flows, transfer flows, and customer queue flows. It is not supported in whisper flows or hold flows.

Corresponding block in the UI

Set callback number

CreateCustomerProfile

Create a customer profile. Customer Profiles must be enabled for your Connect Customer instance.

See CreateProfile in the *Connect Customer Customer Profiles API Reference*.

Parameter object

```
{
"ProfileRequestData": {
    All of these fields are optional.
        "FirstName",
        "MiddleName",
        "LastName",
        "PhoneNumber",
        "EmailAddress",
        "AccountNumber",
        "AdditionalInformation",
        "PartyType",
        "BusinessName",
        "BirthDate",
        "Gender",
        "MobilePhoneNumber",
        "HomePhoneNumber",
        "BusinessPhoneNumber",
        "BusinessEmailAddress",
        "Address1",
        "Address2",
        "Address3",
        "Address4",
```

```
        "City",
        "County",
        "Country",
        "PostalCode",
        "Province",
        "State",
        "ShippingAddress1",
        "ShippingAddress2",
        "ShippingAddress3",
        "ShippingAddress4",
        "ShippingCity",
        "ShippingCounty",
        "ShippingCountry",
        "ShippingPostalCode",
        "ShippingProvince",
        "ShippingState",
        "MailingAddress1",
        "MailingAddress2",
        "MailingAddress3",
        "MailingAddress4",
        "MailingCity",
        "MailingCounty",
        "MailingCountry",
        "MailingPostalCode",
        "MailingProvince",
        "MailingState",
        "BillingAddress1",
        "BillingAddress2",
        "BillingAddress3",
        "BillingAddress4",
        "BillingCity",
        "BillingCounty",
        "BillingCountry",
        "BillingPostalCode",
        "BillingProvince",
        "BillingState",
        "Attributes.x"
    },
   "ProfileResponseData": {
       All of these fields are optional.
       Newly created profile ID is persisted under the Customer -> ProfileID attribute
+ $.Customer.ProfileId
        "FirstName",
        "MiddleName",
```

```
        "LastName",
        "PhoneNumber",
        "EmailAddress",
        "AccountNumber",
        "AdditionalInformation",
        "PartyType",
        "BusinessName",
        "BirthDate",
        "Gender",
        "MobilePhoneNumber",
        "HomePhoneNumber",
        "BusinessPhoneNumber",
        "BusinessEmailAddress",
        "Address1",
        "Address2",
        "Address3",
        "Address4",
        "City",
        "County",
        "Country",
        "PostalCode",
        "Province",
        "State",
        "ShippingAddress1",
        "ShippingAddress2",
        "ShippingAddress3",
        "ShippingAddress4",
        "ShippingCity",
        "ShippingCounty",
        "ShippingCountry",
        "ShippingPostalCode",
        "ShippingProvince",
        "ShippingState",
        "MailingAddress1",
        "MailingAddress2",
        "MailingAddress3",
        "MailingAddress4",
        "MailingCity",
        "MailingCounty",
        "MailingCountry",
        "MailingPostalCode",
        "MailingProvince",
        "MailingState",
        "BillingAddress1",
```

```
        "BillingAddress2",
        "BillingAddress3",
        "BillingAddress4",
        "BillingCity",
        "BillingCounty",
        "BillingCountry",
        "BillingPostalCode",
        "BillingProvince",
        "BillingState",
        "Attributes.x"
}
```

Results and conditions

None. Conditions are not supported. If an error does not occur, the response's attributes are available dynamically under the `$.Customer` path based on the attributes included in `ProfileResponseData`.

Errors

- NoMatchingError - if no other Error matches.

Corresponding block in the UI

Customer profiles block

InvokeLambdaFunction

Invokes an AWS Lambda function with a collection of optional parameters. This AWS Lambda function is also given a copy of the flow run data if there is an associated contact with the flow.

Parameter object

```
{
    "LambdaFunctionARN": The ARN of the AWS Lambda function to be invoked. May be
 defined statically or dynamically.
    "InvocationTimeLimitSeconds": The number of seconds to wait for a response from the
 AWS Lambda function. Must be greater than 0, no larger than 8, and an integer. Must be
 set statically.
    "InvocationType": Specifies the invocation type, allowed values: "SYNCHRONOUS" |
 "ASYNCHRONOUS".
    "LambdaInvocationAttributes" { A map of additional data to send to the AWS Lambda
 function when invoking it. Keys and values may be set statically or dynamically.
```

```
    }
    "ResponseValidation": {
      "ResponseType": "STRING_MAP" or "JSON", If response validation is set to
 STRING_MAP, then the Lambda function should return a flat object of key/value pairs of
 the string type. Otherwise, if response validation is set to JSON, the Lambda function
 can return any valid JSON, including nested JSON. Must be set statically.
    }
}
```

Results and conditions

None. Conditions are not supported. If an error does not occur, the response's attributes are available dynamically under the $.External path.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

None. This action is supported by all channels and in all types of flows.

Corresponding block in the UI

AWS Lambda function

GetCustomerProfile

Retrieve a customer profile based any search identifier, up to five total. Customer Profiles must be enabled for your Connect Customer instance.

See SearchProfiles in the *Connect Customer Customer Profiles API Reference*.

Parameter object

At least one search identifier must be present.

```
{
    "ProfileRequestData": {
        Requires either IdentiferName and IdentifierValue, or SearchCriteria.

        "IdentifierName": Name to search for profiles with one identifier,
        "IdentiferValue": Value to search for profiles with one identifier,
```

```
       "SearchCriteria": [
           {
               "IdentifierName": Name to search for profiles with multiple
identifiers,
               "IdentifierValue": Value to search for profiles with multiple
identifiers
           }
       ],
       Required when using SearchCriteria
       "LogicalOperator": AND or OR

   },
  "ProfileResponseData": {
      All of these fields are optional.
      Newly created profile ID is persisted under the Customer -> ProfileID attribute
+ $.Customer.ProfileId
       "FirstName",
       "MiddleName",
       "LastName",
       "PhoneNumber",
       "EmailAddress",
       "AccountNumber",
       "AdditionalInformation",
       "PartyType",
       "BusinessName",
       "BirthDate",
       "Gender",
       "MobilePhoneNumber",
       "HomePhoneNumber",
       "BusinessPhoneNumber",
       "BusinessEmailAddress",
       "Address1",
       "Address2",
       "Address3",
       "Address4",
       "City",
       "County",
       "Country",
       "PostalCode",
       "Province",
       "State",
       "ShippingAddress1",
       "ShippingAddress2",
       "ShippingAddress3",
```

```
        "ShippingAddress4",
        "ShippingCity",
        "ShippingCounty",
        "ShippingCountry",
        "ShippingPostalCode",
        "ShippingProvince",
        "ShippingState",
        "MailingAddress1",
        "MailingAddress2",
        "MailingAddress3",
        "MailingAddress4",
        "MailingCity",
        "MailingCounty",
        "MailingCountry",
        "MailingPostalCode",
        "MailingProvince",
        "MailingState",
        "BillingAddress1",
        "BillingAddress2",
        "BillingAddress3",
        "BillingAddress4",
        "BillingCity",
        "BillingCounty",
        "BillingCountry",
        "BillingPostalCode",
        "BillingProvince",
        "BillingState",
        "Attributes.x"
}
```

Results and conditions

None. Conditions are not supported. If an error does not occur, the response's attributes are available dynamically under the `$.Customer` path based on the attributes included in `ProfileResponseData`.

Errors

- MultipleFoundError - if multiple profiles were found for the associated profile search key.
- NoneFoundError - if no profiles were found for the associated profile search key.
- NoMatchingError - if no other Error matches.

Corresponding block in the UI

Customer profiles block

GetCustomerProfileObject

Retrieve a customer profile object of the desired type, based on recency or any search identifier. Customer Profiles must be enabled for your Connect Customer instance.

See ListProfileObjects in the *Connect Customer Customer Profiles API Reference*.

Parameter object

A `ProfileId` and `ObjectType` must be present. Either `UseLatest`, or `IdentifierName` and `IdentifierValue` must be present.

```
{
    "ProfileRequestData": {
        "ProfileId": Profile owning the object,
        "ObjectType": Type of object being retrieved,

        "IdentifierName": Optional name of search identifier,
        "IdentifierValue": Optional value of search identifier,
        "UseLatest": true / false
    },
   "ProfileResponseData": {
       All of these fields are optional.
       Asset ID, if a single asset is found, is always persisted under the Customer ->
 Asset -> AssetID attribute + $.Customer.Asset.AssetId
       Order ID, if a single order is found, is always persisted under the Customer ->
 Order -> OrderId attribute + $.Customer.Order.OrderId
       Case ID, if a single case is found, is always persisted under the Customer ->
 Case -> CaseID attribute + $.Customer.Case.CaseId
        "AssetAssetId",
        "AssetProfileId",
        "AssetAssetName",
        "AssetSerialNumber",
        "AssetModelNumber",
        "AssetModelName",
        "AssetProductSKU",
        "AssetPurchaseDate",
        "AssetUsageEndDate",
        "AssetStatus",
        "AssetPrice",
```

```
        "AssetQuantity",
        "AssetDescription",
        "AssetAdditionalInformation",
        "AssetDataSource",
        "AssetAttributes.x",
        "OrderOrderId",
        "OrderProfileId",
        "OrderCustomerEmail",
        "OrderCustomerPhone",
        "OrderCreatedDate",
        "OrderUpdatedDate",
        "OrderProcessedDate",
        "OrderClosedDate",
        "OrderCancelledDate",
        "OrderCancelReason",
        "OrderName",
        "OrderAdditionalInformation",
        "OrderGateway",
        "OrderStatus",
        "OrderStatusCode",
        "OrderStatusUrl",
        "OrderCreditCardNumber",
        "OrderCreditCardCompany",
        "OrderFulfillmentStatus",
        "OrderTotalPrice",
        "OrderTotalTax",
        "OrderTotalDiscounts",
        "OrderTotalItemsPrice",
        "OrderTotalShippingPrice",
        "OrderTotalTipReceived",
        "OrderCurrency",
        "OrderTotalWeight",
        "OrderBillingName",
        "OrderBillingAddress1",
        "OrderBillingAddress2",
        "OrderBillingAddress3",
        "OrderBillingAddress4",
        "OrderBillingCity",
        "OrderBillingCounty",
        "OrderBillingCountry",
        "OrderBillingPostalCode",
        "OrderBillingProvince",
        "OrderBillingState",
        "OrderShippingName",
```

```
        "OrderShippingAddress1",
        "OrderShippingAddress2",
        "OrderShippingAddress3",
        "OrderShippingAddress4",
        "OrderShippingCity",
        "OrderShippingCounty",
        "OrderShippingCountry",
        "OrderShippingPostalCode",
        "OrderShippingProvince",
        "OrderShippingState",
        "OrderAttributes.x",
        "CaseCaseId",
        "CaseProfileId",
        "CaseTitle",
        "CaseSummary",
        "CaseStatus",
        "CaseReason",
        "CaseCreatedBy",
        "CaseCreatedDate",
        "CaseUpdatedDate",
        "CaseClosedDate",
        "CaseAdditionalInformation",
        "CaseDataSource",
        "CaseAttributes.x",
        "ObjectAttributes.x"
    }
}
```

Results and conditions

None. Conditions are not supported. If an error does not occur, the response's attributes are available dynamically under the `$.Customer` path based on the attributes included in `ProfileResponseData`.

Errors

- NoneFoundError - if no profiles were found for the associated profile search key.
- NoMatchingError - if no other Error matches.

Corresponding block in the UI

Customer profiles block

GetCalculatedAttributesForCustomerProfile

Retrieve calculated attributes for a customer profile. Customer Profiles must be enabled for your Connect Customer instance.

Parameter object

A `ProfileId` must be present.

```
{
    "ProfileRequestData": {
        "ProfileId": Profile owning the calculated attribute
    },
   "ProfileResponseData": {
       All of these fields are optional.
       "CalculatedAttributes._average_hold_time",
       "CalculatedAttributes._frequent_caller",
       "CalculatedAttributes.x",
   }
}
```

Results and conditions

None. Conditions are not supported. If an error does not occur, the response's attributes are available dynamically under the `$.Customer` path based on the attributes included in `ProfileResponseData`.

Errors

- NoneFoundError - if no profiles were found for the associated profile search key.
- NoMatchingError - if no other Error matches.

Corresponding block in the UI

Customer profiles block

UpdateCustomerProfile

Update a customer profile that was previously created or retrieved in the flow. Customer Profiles must be enabled for your Connect Customer instance.

See UpdateProfile in the *Connect Customer Customer Profiles API Reference*.

Parameter object

```
{
"ProfileRequestData": {
    All of these fields are optional.
        "FirstName",
        "MiddleName",
        "LastName",
        "PhoneNumber",
        "EmailAddress",
        "AccountNumber",
        "AdditionalInformation",
        "PartyType",
        "BusinessName",
        "BirthDate",
        "Gender",
        "MobilePhoneNumber",
        "HomePhoneNumber",
        "BusinessPhoneNumber",
        "BusinessEmailAddress",
        "Address1",
        "Address2",
        "Address3",
        "Address4",
        "City",
        "County",
        "Country",
        "PostalCode",
        "Province",
        "State",
        "ShippingAddress1",
        "ShippingAddress2",
        "ShippingAddress3",
        "ShippingAddress4",
        "ShippingCity",
        "ShippingCounty",
        "ShippingCountry",
        "ShippingPostalCode",
        "ShippingProvince",
        "ShippingState",
        "MailingAddress1",
        "MailingAddress2",
        "MailingAddress3",
        "MailingAddress4",
```

```
        "MailingCity",
        "MailingCounty",
        "MailingCountry",
        "MailingPostalCode",
        "MailingProvince",
        "MailingState",
        "BillingAddress1",
        "BillingAddress2",
        "BillingAddress3",
        "BillingAddress4",
        "BillingCity",
        "BillingCounty",
        "BillingCountry",
        "BillingPostalCode",
        "BillingProvince",
        "BillingState",
        "Attributes.x"
    },
   "ProfileResponseData": {
       All of these fields are optional.
       Newly created profile ID is persisted under the Customer -> ProfileID attribute
+ $.Customer.ProfileId
        "FirstName",
        "MiddleName",
        "LastName",
        "PhoneNumber",
        "EmailAddress",
        "AccountNumber",
        "AdditionalInformation",
        "PartyType",
        "BusinessName",
        "BirthDate",
        "Gender",
        "MobilePhoneNumber",
        "HomePhoneNumber",
        "BusinessPhoneNumber",
        "BusinessEmailAddress",
        "Address1",
        "Address2",
        "Address3",
        "Address4",
        "City",
        "County",
        "Country",
```

```
        "PostalCode",
        "Province",
        "State",
        "ShippingAddress1",
        "ShippingAddress2",
        "ShippingAddress3",
        "ShippingAddress4",
        "ShippingCity",
        "ShippingCounty",
        "ShippingCountry",
        "ShippingPostalCode",
        "ShippingProvince",
        "ShippingState",
        "MailingAddress1",
        "MailingAddress2",
        "MailingAddress3",
        "MailingAddress4",
        "MailingCity",
        "MailingCounty",
        "MailingCountry",
        "MailingPostalCode",
        "MailingProvince",
        "MailingState",
        "BillingAddress1",
        "BillingAddress2",
        "BillingAddress3",
        "BillingAddress4",
        "BillingCity",
        "BillingCounty",
        "BillingCountry",
        "BillingPostalCode",
        "BillingProvince",
        "BillingState",
        "Attributes.x"
}
```

Results and conditions

None. Conditions are not supported. If an error does not occur, the response's attributes are available dynamically under the `$.Customer` path based on the attributes included in `ProfileResponseData`.

Errors

- NoMatchingError - if no other Error matches.

Corresponding block in the UI

Customer profiles block

Participant actions in the Connect Customer Flow language

Participant actions are attempted only when the flow is run in context of a participant. They generally result in an action that the participant experiences, such as playing a prompt or disconnecting.

Contents

ConnectParticipantWithLexBot

Connects the participant with the specified Amazon Lex bot. When the interaction is over, the Intent and Slots of the bot are available to the flow during its run.

Action syntax

```
{
      "Parameters": {
        "PromptId": "string",
        "Text": "string",
        "SSML": "string",
        "Media": {
            "Uri": "string",
            "SourceType": "string",
            "MediaType": "string"
        },
```

```
        "LexV2Bot": {
          "AliasArn": "string"
        },
        "LexBot": {
            "Name": "string",
            "Region": "string",
            "Alias": "string"
        },
        "LexSessionAttributes": {
          "string": "string"
        },
        "LexInitializationData": {
          "InitialMessage": "string"
        },
        "LexTimeoutSeconds": {
          "Text": "number"
        }
      },
      "Identifier": "string",
      "Type": "ConnectParticipantWithLexBot",
      "Transitions": {
        "NextAction": "string",
        "Errors": [
          {
            "NextAction": "string",
            "ErrorType": "InputTimeLimitExceeded"
          },
          {
            "NextAction": "string",
            "ErrorType": "NoMatchingError"
          },
          {
            "NextAction": "string",
            "ErrorType": "NoMatchingCondition"
          }
        ]
      }
    }
```

Parameter object

Provide either LexBot or LexV2Bot object depending on the Amazon Lex version in the following format.

PromptId

A prompt ID or prompt ARN to play to the participant along with gathering input. This is an optional property and is not required if Text or SSML is specified. This must be either statically defined or a single valid JSONPath identifier.

Type: String

Required: No

Text

An optional string that defines text to send to the participant along with gathering input. May not be specified if PromptId or SSML is also specified. May be defined statically or dynamically.

Type: String

Required: No

SSML

An optional string that defines SSML to send to the participant along with gathering input. May not be specified if Text or PromptId is also specified. May be defined statically or dynamically.

Type: String

Required: No

Media

An optional object that defines an external media source.

- **Uri**: Location of the message.
- **SourceType**: The source from which the message will be fetched. The only supported type is S3.
- **MediaType**: The type of the message to be played. The only supported type is Audio.

LexV2Bot

The details of the LexV2 bot to invoke.

- **AliasArn**: The alias ARN of the LexV2 bot to invoke. May be specified statically or dynamically.

Required: Yes

LexSessionAttributes

A map of session attributes to pass to the Amazon LexV2 bot when it is invoked. The keys and values may be static or dynamic.

Required: No

LexInitializationData

A mapping that defines the initialization data which will be passed to Lex to prime the bot to improve end-customer experience. Lex initialization experience is only supported for Chat channel today. The Voice channel is not supported.

- **InitialMessage**: An optional string that defines the initial message parsed to Lex to prime the bot. The value for InitialMessage parameter is always serialized to$.Media.InitialMessage which resolves to the customer's initial chat message.

Required: No

> **Note**
>
> If the initial message attribute is not included as part of the contact, it will result in the Get customer input block taking the error branch in the flow. To have separate flow configurations for different messaging types, such as web chat, SMS, or Apple Messages for Business, you can use the Check contact attributes block prior to using the Get customer input block to verify the initial message is available.

LexTimeoutSeconds

A mapping that defines the length of Lex timer in second, which will timeout inactive customers in a Lex interaction.

- **Text**: An optional string that defines the Lex timer length for chat.

Required: No

Transitions

If the Amazon Lex interaction succeeds, the result is the Intent of the bot. Conditions are supported, but only the Equals operator is supported within these conditions.

Errors

- **NoMatchingCondition**: If no specified condition evaluated to True.
- **NoMatchingError**: If an error occurred and no other error matched.
- **InputTimeLimitExceeded**: if there is no response before the configured LexTimeoutSeconds.

Conditions

- **NextAction**: A string that contains the Identifier of the Action that should be run after this Action, if no error or condition is preferentially chosen.
- **Supported Conditions**: Conditions are supported but only the EQUALS operator may be used on conditions.

Restrictions

This action is supported by all channels.

This action is available only in contact flows, transfer flows, and customer queue flows. It is not available in whisper flows or hold flows.

Corresponding block in the UI

Get customer input

DisconnectParticipant

Disconnects the participant from the contact and stops this flow from running.

Parameter object

No parameters are expected.

Results and conditions

None. Conditions are not supported.

Errors

None.

Restrictions

This action is supported for all channels and in contact flows, transfer flows, and customer queue flows.

Corresponding block in the UI

Disconnect / hang up

GetParticipantInput

Gathers customer input (a DTMF collection for voice contacts, or an entered string for other channels). There are many optional behaviors after gathering this: encryption, validation, storing to a "LastParticipantInput" section on the flow run data, specifying a custom DTMF terminator for voice contacts and so on. Details are in the parameter object section.

Parameter object

```
{
    "PromptId": [Optional] A prompt ID or prompt ARN to play to the participant along
 with gathering input. May not be specified if Text or SSML is also specified. Must be
 either statically defined or a single valid JSONPath identifier.
    "Text":  An optional string that defines text to send to the participant along with
 gathering input. May not be specified if PromptId or SSML is also specified. May be
 defined statically or dynamically.
    "SSML": An optional string that defines SSML to send to the participant along with
 gathering input. May not be specified if Text or PromptId is also specified. May be
 defined statically or dynamically.
    "Media": { An optional object that defines an external media source
        "Uri": Location of the message
        "SourceType": The source from which the message will be fetched. The only
 supported type is S3
        "MediaType": The type of the message to be played. The only supported type is
 Audio
    }
    "InputTimeLimitSeconds": The number of seconds to wait for input to be collected
 before proceeding with a timeout error. For the Voice channel this is the timeout
 until the *first* DTMF digit is entered. Must be defined statically, and must be a
 valid integer larger than zero.
    "StoreInput": "True" or "False". Must be statically defined.
```

```
   "InputValidation": { An object that defines how to validate customer inputs,
 required if and only if StoreInput is True
       "PhoneNumberValidation": { Optional, one of the ways to validate inputs, make
 sure that it's a valid phone number. May not be specified if CustomValidation is
 specified.
            "NumberFormat": "Local" or "E164". If "Local" is specified, it is
 validated to be a local number (without the + and the country code), "E164" enforces
 that the customer input is a fully defined e.164 phone number. Must be defined
 statically.
            "CountryCode": If the number format is "Local", this must be defined. This
 is the two letter country code to be associated with the input number when validating.
 Must be defined statically.
       }
       "CustomValidation": { Optional, the other way to validate inputs. May not be
 specified if PhoneNumberValidation is specified.
           "MaximumLength": A number representing the maximum length of the input. May
 be defined statically or dynamically.
       }
   },
   "InputEncryption": { An optional object that defines how to encrypt the customer
 input. May only be specified if "CustomValidation" is provided.
       "EncryptionKeyId": The identifier of a key that has been uploaded in the AWS
 console for the purposes of customer input encryption. May be specified statically or
 dynamically.
       "Key": The PEM definition of the public key to use to encrypt this data. This
 key must be signed with the encryption key identified by the EncryptionKeyId. May be
 specified statically or dynamically.
   },
   "DTMFConfiguration": { An optional object to override default DTMF behavior for
 voice calls
       "InputTerminationSequence": Up to five digits to serve as the terminating
 sequence when gathering DTMF
       "DisableCancelKey": "True" or "False". If "True", the "*" key doesn't cancel
 gathering DTMF digits.
       "InterdigitTimeLimitSeconds": The number of seconds to wait between each DTMF
 digit entry before proceeding with a timeout error. This timeout applies after the
 first digit has been entered and resets after each subsequent digit. May be defined
 statically or dynamically, and must be a valid integer between 1 and 20 seconds.
   }
}
```

Results and conditions

If the "StoreInput" option is "True", there is no run result and conditions are not supported. If the "StoreInput" option is not defined or is "False", the run result is the participant input, and conditions are supported but only the Equals operator may be used on conditions. The values being compared must be static and be a single character - 0-9 numeric, *, or #.

Errors

- NoMatchingCondition - None of the specified conditions evaluated to true. Must be defined only if StoreInput is False.
- NoMatchingError - if no other Error matches. Must always be defined.
- InvalidPhoneNumber - the stored input was not a valid phone number according to the specified PhoneNumberValidation. Must be defined only if StoreInput is true, and PhoneNumberValidation is specified.
- InputTimeLimitExceeded - if there is no response before the configured InputTimeLimitSeconds.

Restrictions

This action is only supported on the voice channel.

This action can be used in contact flows, transfer flows, and customer queue flows but not in whisper flows or hold flows.

Corresponding block in the UI

Get customer input

MessageParticipant

Sends a message to the participant. This is an audio prompt or text-to-speech for voice contacts, or a text message for other channels.

Parameter object

```
{
    "PromptId": [Optional] A prompt ID or prompt ARN to play to the participant along
 with gathering input. May not be specified if Text or SSML is also specified. Must be
 specified either statically or as a single valid JSONPath identifier.
```

```
    "Text":  An optional string that defines text to send to the participant along with
 gathering input. May not be specified if PromptId or SSML is also specified. May be
 specified statically or dynamically.
    "SSML": An optional string that defines SSML to send to the participant along with
 gathering input. May not be specified if Text or PromptId is also specified May be
 specified statically or dynamically.
    "Media": { An optional object that defines an external media source
        "Uri": Location of the message
        "SourceType": The source from which the message will be fetched. The only
 supported type is S3
        "MediaType": The type of the message to be played. The only supported type is
 Audio
    }
}
```

Results and conditions

None. No conditions are supported.

Errors

NoMatchingError - If an error occurred and no other error matched.

Restrictions

This action is supported in contact flows, transfer flows, whisper flows, and customer queue flows. It is not supported in hold flows.

"PromptId" and "SSML" are only supported for the voice channel. All other channels support only the "Text" option.

Corresponding block in the UI

Play

MessageParticipantIteratively

Loops a sequence of prompts while a customer or agent is on hold or in queue. This block can be configured with an interruption timeout when in a Queue flow that interrupts the message loop to run other flow logic. The message loop can include entries for both Text and Prompts.

Parameter object

```
{
```

```
  "Messages" : [ A List of messages to be played in a loop. These are defined with
 either TTS or a Prompt
      {
        "Text" : An optional string that defines text to send to the participant
      },
      {
        "PromptId" : A prompt ID or prompt ARN to play to the participant
      },
      {
        "SSML" : An optional string that defines the ssml
      },
      {
        "Media": { An optional object that defines an external media source
          "Uri": Location of the message
          "SourceType": The source from which the message will be fetched. The only
 supported type is S3
          "MediaType": The type of the message to be played. The only supported type
 is Audio
        }
      }

  ],
  "InterruptFrequencySeconds" : [Optional] Time to elapse before the action completes
 with "MessagesInterrupted" run result
}
```

Results and conditions

When the timeout elapses, the action completes with the result as "MessagesInterrupted". Conditions are supported, but only the "Equals" operator is supported. The only supported operand is MessagesInterrupted.

Errors

- NoMatchingError - if no other Error matches.

Restrictions

This action is supported in Customer Queue, Customer Hold, and Agent Hold flows.

"PromptId" is supported only for the Voice channel, all other channels support only the "Text" option.

If this action is used on the chat channel, it immediately takes the error branch. If no error branch is available, the flow stop running and the contact is routed to next available agent.

Corresponding block in the UI

Loop prompt

ShowView

Initiates a UI-based workflow that can be surfaced to users of front end applications. This action can be used to create step-by-step guides for agents who are using the Connect Customer agent workspace.

Parameter object

```
{
    "ViewResource": {
        "Id": "Id of the View Resource that will be shown in the UI.",
        "Version": "Version of the View Resource that will be shown in the UI."
    },
    "InvocationTimeLimitSeconds": 400,
    "ViewData": {
        "Description": "An optional map of data that will be passed to the View
 Resource. Keys and values may be set statically or dynamically."
    },
    "SensitiveDataConfiguration": {
        "HideResponseOn": ["TRANSCRIPT"]
    }
}
```

Results and conditions

The result that the user selects when interacting with the View. The available conditions will be dependent on the View resource specified in action parameters.

Errors

- NoMatchingError - if no other Error matches.
- NoMatchingCondition - if no other Condition matches.

- TimeLimitExceeded - if there is no response before the configured `InvocationTimeLimitSeconds`.

Restrictions

This action is only supported on the chat channel.

This action can be used in inbound flows and customer queue flows.

To ensure reliable show view rendering, limit combined inputs and contact attributes to 16KB or less.

Corresponding block in the UI

Show View

This action routes step-by-step guides that are to be displayed to agents in the agent workspace. It routes the guides as chat contacts. This type of chat contact is different from the customer-based contact that the agent is handling.

This action can only be used in inbound contact flows.

Connect Customer Rules Function language

This section describes the Connect Customer Rules Function language. You can use it to add conditions to rules programmatically. The function language is a JSON-based representation of a series of rule conditions.

Contents

Concepts

The following terms are used in the Rules Function language.

Operator

A function that is used to evaluate Operands. The very top Operator has to be either AND or OR.

Operands

An array of objects that Operator is evaluating on. The length the array and the type of each object depends on the Operator defined on the same level.

ComparisonValue

A JSON path string that specifies the value field the rule is comparing.

FilterClause

An object that defines additional criteria that the rule is evaluating against. Depending on **ComparisonValue**, the value of this field varies.

Negate

A Boolean value that indicates if negation should be applied to the operator.

Example rule function in Connect Customer Rule Function language

The following example shows a simple rule that would be evaluated to `true` if "refund" is mentioned by the customer who is in a specific queue.

To learn how to use conditions correctly, we recommend creating a new rule in the Connect Customer console, and then calling the DescribeRule API for it.

```
{
  "Version": "2022-11-25", // A string representing the version of the rule. Currently
 the only supported version is 2022-11-25.
  "RuleFunction": { // RuleFunction object body
    "Operator": "AND",
    "Operands": [
      {
        "Operator": "CONTAINS_ANY",
        "Operands": [
          "refund"
        ],
        "ComparisonValue": "$.ContactLens.RealTimeCall.ExactMatch.Transcript",
        "FilterClause": {
          "LogicOperator": "AND",
          "Filters": [
            {
              "Type": "ParticipantRole",
              "Data": "CUSTOMER"
            }
          ]
        },
        "Negate": false // If Negate were set to true, it would mean the word "refund"
 was NOT mentioned
      },
      {
        "Operator": "CONTAINS_ANY",
        "Operands": [
          "11111111-1234-5678-9123-12345678012" // QueueId
        ],
        "ComparisonValue": "$.ContactLens.RealTimeCall.Queue.QueueId",
        "Negate": false
      }
    ]
```

```
  }
}
```

Conditions in Connect Customer Rules Function language

A rule function needs to start with either an AND or OR Operator. The Operands of AND or OR Operators is a list of conditions. Conditions vary depending on the TriggerEventSource.

Following are the conditions that you can use.

Contents

OnMetricDataUpdate

Metric Data Update - Rule Conditions

Parameters

- Operator - "AND" or "OR"
- Operands – An array of `MetricGroup`, can support up to two groups.
- FilterClause – Rule level metric filter for **End Dimension** or grouping. Choosing an End Dimension restricts the usage of metrics, as described in Metric Data Update - MetricCondition.

```
"Operator": "AND", // "AND or "OR"
"Operand": MetricGroup[], // Upto 2 array elements
"FilterClause" {
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "MetricDataGrouping",
            "Data": ["AGENT" | "QUEUE" | "ROUTING_PROFILE" | "FLOW"] // // Only one is
 supported
        },
    ]
}
```

Metric Data Update - MetricGroup

Each MetricGroup can have up to compatible 10 MetricConditions grouped by an "AND" or "OR". Up to two MetricGroups are supported within the same Rule Condition.

```
"Operands": [
    {
        "Operator": "AND", // "AND or "OR"
        "Operands": MetricCondition[], // Upto 10 array elements
    },
    ...
]
```

Metric Data Update - Metric Filters

Every MetricCondition has a compound filter format, and requires a list of mandatory filters and some optional filters.

Required filters

These filters are always required in MetricCondition for the when supported, but not supported in others.

MetricDataPeriodSeconds

The trailing time period window, in seconds, for which the metric is calculated. All MetricCondition within the same MetricGroup are required to have a consistent MetricDataPeriodSeconds, that is, the exact same filter in all of them, or none of them should have these filters. The time window has to less than 24 hours (< 86400).

```
{
    "Type": "MetricDataPeriodSeconds", // Required for Historical mertcis
    "Data": { "Past": 300}
}
```

Supported metrics: AVG_HANDLE_TIME, AVG_QUEUE_ANSWER_TIME, AVG_INTERACTION_TIME, AVG_HOLD_TIME, SERVICE_LEVEL_X, AGENT_OCCUPANCY. These metrics cannot be clubbed with other metrics in the same MetricGroup.

AgentActivityDurationSeconds

This filter is only required for AGENT_ACTIVITY metric, and determines the time period window, in seconds, for which the Agent Activity status is held. It has to less than 24 hours (< 86400).

```
{
    "Type": "AgentActvityDurationSeconds",
    "Data": { "Past": 300}
}
```

MetricDataThresholdSeconds

This filter is only required for SERVICE_LEVEL_X, and is defined here.

```
{
    "Type": "MetricDataThresholdSeconds",
    "Data": { "Comparison": "LT", "ThresholdValue": 300} // Comparison == "LT" only
}
```

End Dimension filters

Exactly one of the below filters is required for every MetricCondition. If the MetricConditions are in the same group, then they are required to have the same End Dimension Filter. For more information about these filters, see GetMetricDataV2 request filters.

MetricDataFilterByAgent

Provide the list of Agent IDs in MetricCondition. Up to 25 values can be provided.

```
{
    "Type": "MetricDataFilterByAgent",
```

```
    "Data": ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"]
}
```

MetricDataFilterByAgentHierarchyLevelX

Agent Hierarchy filters can be any of the five types:

1. MetricDataFilterByAgentHierarchyLevelOne
2. MetricDataFilterByAgentHierarchyLevelTwo
3. MetricDataFilterByAgentHierarchyLevelThree
4. MetricDataFilterByAgentHierarchyLevelFour
5. MetricDataFilterByAgentHierarchyLevelFive

Provide the Agent Hierarchy ID in the filter.

```
{
    "Type": "MetricDataFilterByAgentHierarchyLevelFive",
    "Data": ["b99ba702-b84b-4279-abad-5ceda9ff2640"] // One array element only
}
```

> **Note**
>
> Only one value is supported in the filter.

MetricDataFilterByQueue

Provide the list of Queue IDs in MetricCondition. Up to 25 values can be provided.

```
{
    "Type": "MetricDataFilterByQueue",
    "Data": ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"]
}
```

MetricDataFilterByRoutingProfile

Provide the list of Routing Profile IDs in MetricCondition. Up to 25 values can be provided.

```
{
    "Type": "MetricDataFilterByRoutingProfile",
    "Data": ["b653beae-0a34-4f0c-83d3-067c9b8370bf" , "60a4c89a-
b485-4bf2-8396-2a151b4e05a7"]
}
```

MetricDataFilterByFlow

Provide the list of Flow IDs in MetricCondition. Up to 25 values can be provided.

```
{
    "Type": "MetricDataFilterByFlow",
    "Data": ["a04377e2-zb01-g182-nc02-a285a66279en", "ae5a6965-zb64-g20d-n8a8-
a83d2a15081n"]
}
```

MetricDataFilterByFlowModule

Provide the list of flow module IDs in MetricCondition. Up to 25 values can be provided.

```
{
    "Type": "MetricDataFilterByFlowModule",
    "Data": ["a04377e2-zb01-g182-nc02-a285a66279en", "ae5a6965-zb64-g20d-n8a8-
a83d2a15081n"]
}
```

MetricDataFilterByFlowOutcomeType

Flow Outcome Type filter values can be any of the following depending on the filter resource is flow/module.

- DROPPED
- DISCONNECTED_PARTICIPANT
- ENDED_FLOW_EXECUTION
- TRANSFERRED_TO_QUEUE
- TRANSFERRED_TO_AGENT
- TRANSFERRED_TO_FLOW
- TRANSFERRED_TO_PHONE_NUMBER
- RETURNED_TO_FLOW

```
{
    "Type": "MetricDataFilterByFlowOutcomeType",
    "Data": ["DROPPED", "TRANSFERRED_TO_QUEUE"] // Multiple values can be provided
}
```

Optional filters

The following filters are optional in MetricCondition.

MetricDataFilterByChannel

This is an optional filter that is supported in MetricCondition for all metrics. However, if used, it is require to be consistent for every MetricCondition.

```
{
    "Type": "MetricDataFilterByChannel",
    "Data": ["VOICE", "CHAT", "TASK"], // any subset of this is valid
}
```

Exception: It cannot be used with AGENT_ACTIVITY and AGENT_OCCUPANCY metrics, but can be used for other metrics in the same RuleCondition but within a different MetricGroup.

Metric Data Update - MetricCondition

Contents

- AGENT_ACTIVITY
- OLDEST_CONTACT_AGE
- AGENTS_AVAILABLE
- CONTACTS_IN_QUEUE
- AVG_QUEUE_ANSWER_TIME
- AVG_INTERACTION_TIME
- AVG_HOLD_TIME
- AVG_HANDLE_TIME
- AGENT_OCCUPANCY
- SERVICE_LEVEL
- FLOWS_STARTED

- AVG_FLOW_TIME
- MAX_FLOW_TIME
- MIN_FLOW_TIME
- FLOWS_OUTCOME
- PERCENT_FLOWS_OUTCOME

AGENT_ACTIVITY

For more information about the AGENT_ACTIVITY metric, see AGENT_ACTIVITY and About agent status.

The `AGENT_ACTIVITY` metric supports an agent `statusId`, or string values of one of the following states: `ERROR`, `MISSED`, `REJECTED`, `ON_CONTACT`, `AFTER_CONTACT_WORK`, `INCOMING`, `AVAILABLE`.

The following example shows:

```
Rule: When any agent with Hierarchy Level 1 (filterValue = b99ba702-b84b-4279-
abad-5ceda9ff2640) has
                        had AGENT_ACTIVITY equal to statusId ee8c71b9-49c2-4d00-
ab6d-244f2d9a5c58 for the last 600 seconds, then perform a RuleAction.
```

```
{
    Operator: "EQUALS", // Fixed value
    ComparisonValue: "$.MetricData.AGENT_ACTIVITY",
    Operands: ["ee8c71b9-49c2-4d00-ab6d-244f2d9a5c58"],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByAgentHierarchyLevelOne",
          Data: ["b99ba702-b84b-4279-abad-5ceda9ff2640"],
        },
        {
          Type: "AgentActivityDurationSeconds",
          Data: { Past: 600 },
        },
      ],
    },
    Negate: true, // or false
```

```
  },
```

Supported End Dimensions (MetricDataGrouping): AGENT

Required Filters: AgentActivityDurationSeconds

Supported End Dimension Filters: MetricDataFilterByAgent, MetricDataFilterByAgentHierarchyLevelX, MetricDataFilterByQueue, MetricDataFilterByRoutingProfile

OLDEST_CONTACT_AGE

For more information about the OLDEST_CONTACT_AGE metric, see GetCurrentMetricData and Oldest.

Rule: OLDEST_CONTACT_AGE metric value, filtered by listed queues and channels ≥ 900 (15 minutes), then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.OLDEST_CONTACT_AGE",
    Operands: [900],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByQueue",
          Data: ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"],
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
      ],
    },
    Negate: false, // Fixed value
}
```

Supported End Dimensions (MetricDataGrouping): QUEUE

Required filters: MetricDataFilterByQueue, MetricDataFilterByRoutingProfile

Supported optional filters: `MetricDataFilterByChannel`

AGENTS_AVAILABLE

For more information about the AGENTS_AVAILABLE metric, see GetCurrentMetricData and Available.

Rule: `AGENTS_AVAILABLE` metric value, filtered by listed queues and channels ≤ 5, then perform a RuleAction.

```
{
    Operator: "NumberLessOrEqualTo", // also supports "NumberGreaterOrEqualTo"
    ComparisonValue: "$.MetricData.AGENTS_AVAILABLE",
    Operands: [5],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByQueue",
          Data: ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"],
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
      ],
    },
    Negate: false, // fixed value
  },
```

Supported End Dimensions (MetricDataGrouping): `QUEUE`, `ROUTING_PROFILE`

Supported End Dimension filters: `MetricDataFilterByQueue`, `MetricDataFilterByRoutingProfile`

Supported optional filters: `MetricDataFilterByChannel`

CONTACTS_IN_QUEUE

For more information about the `CONTACTS_IN_QUEUE` metric, see GetCurrentMetricData and In queue.

Rule: CONTACTS_IN_QUEUE metric value, filtered by listed queues and channels ≥ 100, then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.CONTACTS_IN_QUEUE",
    Operands: [100],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByQueue",
          Data: ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"],
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
      ],
    },
    Negate: false, // fixed value
},
```

Supported End Dimensions (MetricDataGrouping): QUEUE

Supported End Dimension filters: `MetricDataFilterByQueue`, `MetricDataFilterByRoutingProfile`

Supported optional filters: `MetricDataFilterByChannel`

AVG_QUEUE_ANSWER_TIME

For more information about the AVG_QUEUE_ANSWER_TIME metric, see GetCurrentMetricData and Avg queue answer time.

Rule: AVG_QUEUE_ANSWER_TIME metric value, over the given duration of 28800 seconds (8 hours), filtered by listed queues and channels ≥ 900 (15 minutes), then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.AVG_QUEUE_ANSWER_TIME",
```

```
    Operands: [900],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByQueue",
          Data: ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"],
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
        {
          Type: "MetricDataDurationSeconds",
          Data: { Past: 28800 },
        },
      ],
    },
    Negate: false, // fixed value
  }
```

Supported End Dimensions (MetricDataGrouping): QUEUE

Required Filters: `MetricDataDurationSeconds`

Supported End Dimension filters: `MetricDataFilterByQueue`, `MetricDataFilterByRoutingProfile`

Supported optional filters: `MetricDataFilterByChannel`

AVG_INTERACTION_TIME

For more information about the AVG_INTERACTION_TIME metric, see GetMetricDataV2.

Rule: AVG_INTERACTION_TIME metric value, over the given duration of 28800 seconds (8 hours), filtered by listed queues and channels ≥ 1800 (30 minutes), then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.AVG_INTERACTION_TIME",
    Operands: [1800],
    FilterClause: {
```

```
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByQueue",
          Data: ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"],
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
        {
          Type: "MetricDataDurationSeconds",
          Data: { Past: 28800 }, // 8 hours
        },
      ],
    },
    Negate: false, // fixed value
  },
```

Supported End Dimensions (MetricDataGrouping): QUEUE

Required Filters: `MetricDataDurationSeconds`

Supported End Dimension filters: `MetricDataFilterByQueue`, `MetricDataFilterByRoutingProfile`

Supported optional filters: `MetricDataFilterByChannel`

AVG_HOLD_TIME

For more information about the AVG_HOLD_TIME metric, see GetMetricDataV2.

Rule: AVG_HOLD_TIME metric value, over the given duration of 28800 seconds (8 hours), filtered by listed queues and channels ≥ 1800 (30 minutes), then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.AVG_HOLD_TIME",
    Operands: [1800],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
```

```
        {
          Type: "MetricDataFilterByQueue",
          Data: ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"],
        },
        {
          Type:"MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
        {
          Type: "MetricDataDurationSeconds",
          Data: { Past: 28800 },
        },
      ],
    },
    Negate: false, // fixed value
  },
```

Supported End Dimensions (MetricDataGrouping): QUEUE

Required Filters: `MetricDataDurationSeconds`

Supported End Dimension filters: `MetricDataFilterByQueue`, `MetricDataFilterByRoutingProfile`

Supported optional filters: `MetricDataFilterByChannel`

AVG_HANDLE_TIME

For more information about the AVG_HANDLE_TIME metric, see GetMetricDataV2.

Rule: AVG_HANDLE_TIME metric value, over the given duration of 28800 seconds (8 hours), filtered by listed queues and channels ≥ 1800 (30 minutes), then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.AVG_HANDLE_TIME",
    Operands: [1800],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByQueue",
```

```
          Data: ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"],
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
        {
          Type: "MetricDataDurationSeconds",
          Data: { Past: 28800 },
        },
      ],
    },
    Negate: false, // fixed value
  },
```

Supported End Dimensions (MetricDataGrouping): QUEUE, AGENT

Required Filters: `MetricDataDurationSeconds`

Supported End Dimension filters: For QUEUE end dimension, `MetricDataFilterByQueue` and `MetricDataFilterByRoutingProfile` are supported. For AGENT end dimensions, `MetricDataFilterByQueue`, `MetricDataFilterByRoutingProfile`, `MetricDataFilterByAgent`, and `MetricDataFilterByAgentLevelX` are supported

Supported optional filters: `MetricDataFilterByChannel`

AGENT_OCCUPANCY

For more information about the AGENT_OCCUPANCY metric, see GetMetricDataV2.

Rule: AGENT_OCCUPANCY metric value, over the given duration of 28800 seconds (8 hours), filtered by listed queues and channels ≥ 80%, then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.AGENT_OCCUPANCY",
    Operands: [80],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByAgent",
```

```
          Data: ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"],
        },
        {
          Type: "MetricDataDurationSeconds",
          Data: { Past: 300 },
        },
      ],
    },
    Negate: false, // fixed value
  },
```

Supported End Dimensions (MetricDataGrouping): `AGENT`

Required Filters: `MetricDataDurationSeconds`

Supported End Dimension filters: `MetricDataFilterByAgent`, `MetricDataFilterByAgentLevelX`, `MetricDataFilterByRoutingProfile`

SERVICE_LEVEL

For more information about the SERVICE_LEVEL metric, see GetMetricDataV2.

Rule: SERVICE_LEVEL metric value, over the given duration of 600 (10 minutes) metric value, over the given duration of 28800 seconds (8 hours), filtered by listed queues and channels ≤ 50%, then perform a RuleAction.

```
{
    Operator: "NumberLessOrEqualTo", // also supports "NumberGreaterOrEqualTo"
    ComparisonValue: "$.MetricData.SERVICE_LEVEL",
    Operands: [50],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByQueue",
          Data: ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"],
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
```

```
        {
          Type: "MetricDataDurationSeconds",
          Data: { Past: 28800 },
        },
        {
          Type: "MetricDataThresholdSeconds",
          Data: { Comparison: MetricThresholdComparison.LT, ThresholdValue: 600},
        },
      ],
    },
    Negate: false, // fixed value
  },
```

Supported End Dimensions (MetricDataGrouping): QUEUE

Required filters: `MetricDataDurationSeconds`, `MetricDataThresholdSeconds`

Supported End Dimension filters: `MetricDataFilterByQueue`, `MetricDataFilterByRoutingProfile`

Supported optional filters: `MetricDataFilterByChannel`

FLOWS_STARTED

For more information about the FLOWS_STARTED metric, see GetMetricDataV2.

Rule : FLOWS_STARTED metric value, over the given duration of 300 seconds (5 minutes), filtered by listed flows and channels 10, then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.FLOWS_STARTED",
    Operands: [10],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByFlow",
          Data: ["c04377e2-db01-4182-bc02-4285a66279e7", "1e5a6965-7b64-420d-88a8-
a83d2a15081d"],
        },
        {
          Type: "MetricDataDurationSeconds",
```

```
          Data: { Past: 300 }
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
      ],
    },
    Negate: false, // Fixed value
}
```

Supported End Dimensions (MetricDataGrouping): `FLOW`

Required Filters: `MetricDataDurationSeconds`

Supported End Dimension filters: `MetricDataFilterByFlow`, `MetricDataFilterByFlowModule`

Supported optional filters: `MetricDataFilterByChannel`

AVG_FLOW_TIME

For more information about the AVG_FLOW_TIME metric, see GetMetricDataV2.

Rule : `AVG_FLOW_TIME` metric value, over the given duration of 300 seconds (5 minutes), filtered by listed flows, flow outcomes and channels >= 600 (10 minutes), then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.AVG_FLOW_TIME",
    Operands: [600],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByFlow",
          Data: ["5f873afb-f359-4c54-b626-ecad1bd3f774"],
        },
        {
          Type: "MetricDataFilterByFlowOutcomeType",
          Data: ["DROPPED", "TRANSFERRED_TO_QUEUE"],
        },
```

```
        {
          Type: "MetricDataDurationSeconds",
          Data: {
            "Past": 300
          },
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
      ],
    },
    Negate: false, // Fixed value
}
```

Supported End Dimensions (MetricDataGrouping): `FLOW`

Required Filters: `MetricDataDurationSeconds`

Supported End Dimension filters: `MetricDataFilterByFlow`, `MetricDataFilterByFlowModule`, `MetricDataFilterByFlowOutcomeType`

Supported optional filters: `MetricDataFilterByChannel`

MAX_FLOW_TIME

For more information about the MAX_FLOW_TIME metric, see GetMetricDataV2.

Rule : `MAX_FLOW_TIME` metric value, over the given duration of 300 seconds (5 minutes), filtered by listed flows, flow outcomes and channels >= 600 (10 minutes), then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.MAX_FLOW_TIME",
    Operands: [600],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByFlow",
          Data: ["5f873afb-f359-4c54-b626-ecad1bd3f774"],
        },
        {
```

```
          Type: "MetricDataFilterByFlowOutcomeType",
          Data: ["DROPPED", "TRANSFERRED_TO_QUEUE"],
        },
        {
          Type: "MetricDataDurationSeconds",
          Data: {
            "Past": 300
          },
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
      ],
    },
    Negate: false, // Fixed value
}
```

Supported End Dimensions (MetricDataGrouping): `FLOW`

Required Filters: `MetricDataDurationSeconds`

Supported End Dimension filters: `MetricDataFilterByFlow`, `MetricDataFilterByFlowModule`, `MetricDataFilterByFlowOutcomeType`

Supported optional filters: `MetricDataFilterByChannel`

MIN_FLOW_TIME

For more information about the MIN_FLOW_TIME metric, see GetMetricDataV2.

Rule : `MIN_FLOW_TIME` metric value, over the given duration of 300 seconds (5 minutes), filtered by listed flows, flow outcomes and channels >= 600 (10 minutes), then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.MIN_FLOW_TIME",
    Operands: [600],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByFlow",
```

```
          Data: ["5f873afb-f359-4c54-b626-ecad1bd3f774"],
        },
        {
          Type: "MetricDataFilterByFlowOutcomeType",
          Data: ["DROPPED", "TRANSFERRED_TO_QUEUE"],
        },
        {
          Type: "MetricDataDurationSeconds",
          Data: {
            "Past": 300
          },
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
      ],
    },
    Negate: false, // Fixed value
}
```

Supported End Dimensions (MetricDataGrouping): `FLOW`

Required Filters: `MetricDataDurationSeconds`

Supported End Dimension filters: `MetricDataFilterByFlow`, `MetricDataFilterByFlowModule`, `MetricDataFilterByFlowOutcomeType`

Supported optional filters: `MetricDataFilterByChannel`

FLOWS_OUTCOME

For more information about the FLOWS_OUTCOME metric, see GetMetricDataV2.

Rule : `FLOWS_OUTCOME` metric value, over the given duration of 300 seconds (5 minutes), filtered by listed flows, flow outcomes and channels >= 10, then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
    ComparisonValue: "$.MetricData.FLOWS_OUTCOME",
    Operands: [10],
    FilterClause: {
```

```
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByFlow",
          Data: ["5f873afb-f359-4c54-b626-ecad1bd3f774"],
        },
        {
          Type: "MetricDataFilterByFlowOutcomeType",
          Data: ["DROPPED", "TRANSFERRED_TO_QUEUE"],
        },
        {
          Type: "MetricDataDurationSeconds",
          Data: {
            "Past": 300
          },
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
      ],
    },
    Negate: false, // Fixed value
}
```

Supported End Dimensions (MetricDataGrouping): `FLOW`

Required Filters: `MetricDataDurationSeconds`

Supported End Dimension filters: `MetricDataFilterByFlow`, `MetricDataFilterByFlowModule`, `MetricDataFilterByFlowOutcomeType`

Supported optional filters: `MetricDataFilterByChannel`

PERCENT_FLOWS_OUTCOME

For more information about the PERCENT_FLOWS_OUTCOME metric, see GetMetricDataV2.

Rule : `PERCENT_FLOWS_OUTCOME` metric value, over the given duration of 300 seconds (5 minutes), filtered by listed flows, flow outcomes and channels >= 50%, then perform a RuleAction.

```
{
    Operator: "NumberGreaterOrEqualTo", // also supports "NumberLessOrEqualTo"
```

```
    ComparisonValue: "$.MetricData.PERCENT_FLOWS_OUTCOME",
    Operands: [50],
    FilterClause: {
      LogicOperator: "AND", // Always "AND"
      Filters: [
        {
          Type: "MetricDataFilterByFlow",
          Data: ["5f873afb-f359-4c54-b626-ecad1bd3f774"],
        },
        {
          Type: "MetricDataFilterByFlowOutcomeType",
          Data: ["DROPPED", "TRANSFERRED_TO_QUEUE"],
        },
        {
          Type: "MetricDataDurationSeconds",
          Data: {
            "Past": 300
          },
        },
        {
          Type: "MetricDataFilterByChannel",
          Data: ["CHAT"],
        },
      ],
    },
    Negate: false, // Fixed value
}
```

Supported End Dimensions (MetricDataGrouping): `FLOW`

Required Filters: `MetricDataDurationSeconds`

Supported End Dimension filters: `MetricDataFilterByFlow`, `MetricDataFilterByFlowModule`, `MetricDataFilterByFlowOutcomeType`

Supported optional filters: `MetricDataFilterByChannel`

Full examples

Example 1

A rule with `AGENTS_AVAILABLE`, `OLDEST_CONTACT_AGE`, `CONTACTS_IN_QUEUE` in MetricGroup 1 and `AVG_QUEUE_ANSWER_TIME`, `AVG_HOLD_TIME`, `AVG_INTERACTION_TIME`,

AVG_HANDLE_TIME, and SERVICE_LEVEL in Metric Group 2, both of which have End Dimension = QUEUE.

```
{
    "Operator": "OR",
    "Operands": [{
        "Operator": "AND",
        "Operands": [{
            "Operator": "NumberGreaterOrEqualTo",
            "ComparisonValue": "$.MetricData.AGENTS_AVAILABLE",
            "Operands": [0],
            "FilterClause": {
                "LogicOperator": "AND",
                "Filters": [{
                    "Type": "MetricDataFilterByQueue",
                    "Data": ["e7a11ddd-041d-449a-ab1e-6a493cb83626", "26b9c7f4-
ed5c-4cd1-b751-2cbfb4c54ea2"]
                }, {
                    "Type": "MetricDataFilterByChannel",
                    "Data": ["VOICE"]
                }]
            },
            "Negate": false
        }, {
            "Operator": "NumberGreaterOrEqualTo",
            "ComparisonValue": "$.MetricData.CONTACTS_IN_QUEUE",
            "Operands": [0],
            "FilterClause": {
                "LogicOperator": "AND",
                "Filters": [{
                    "Type": "MetricDataFilterByQueue",
                    "Data": ["e7a11ddd-041d-449a-ab1e-6a493cb83626", "26b9c7f4-
ed5c-4cd1-b751-2cbfb4c54ea2"]
                }, {
                    "Type": "MetricDataFilterByChannel",
                    "Data": ["VOICE"]
                }]
            },
            "Negate": false
        }, {
            "Operator": "NumberGreaterOrEqualTo",
            "ComparisonValue": "$.MetricData.OLDEST_CONTACT_AGE",
            "Operands": [0],
```

```
            "FilterClause": {
                "LogicOperator": "AND",
                "Filters": [{
                    "Type": "MetricDataFilterByQueue",
                    "Data": ["e7a11ddd-041d-449a-ab1e-6a493cb83626", "26b9c7f4-
ed5c-4cd1-b751-2cbfb4c54ea2"]
                }, {
                    "Type": "MetricDataFilterByChannel",
                    "Data": ["VOICE"]
                }]
            },
            "Negate": false
        }]
    }, {
        "Operator": "OR",
        "Operands": [{
            "Operator": "NumberGreaterOrEqualTo",
            "ComparisonValue": "$.MetricData.AVG_HANDLE_TIME",
            "Operands": [0],
            "FilterClause": {
                "LogicOperator": "AND",
                "Filters": [{
                    "Type": "MetricDataFilterByQueue",
                    "Data": ["e7a11ddd-041d-449a-ab1e-6a493cb83626", "26b9c7f4-
ed5c-4cd1-b751-2cbfb4c54ea2"]
                }, {
                    "Type": "MetricDataDurationSeconds",
                    "Data": {
                        "Past": 300
                    }
                }, {
                    "Type": "MetricDataFilterByChannel",
                    "Data": ["VOICE"]
                }]
            },
            "Negate": false
        }, {
            "Operator": "NumberGreaterOrEqualTo",
            "ComparisonValue": "$.MetricData.AVG_QUEUE_ANSWER_TIME",
            "Operands": [0],
            "FilterClause": {
                "LogicOperator": "AND",
                "Filters": [{
                    "Type": "MetricDataFilterByQueue",
```

```
                    "Data": ["e7a11ddd-041d-449a-ab1e-6a493cb83626", "26b9c7f4-
ed5c-4cd1-b751-2cbfb4c54ea2"]
                }, {
                    "Type": "MetricDataDurationSeconds",
                    "Data": {
                        "Past": 300
                    }
                }, {
                    "Type": "MetricDataFilterByChannel",
                    "Data": ["VOICE"]
                }]
            },
            "Negate": false
        }, {
            "Operator": "NumberGreaterOrEqualTo",
            "ComparisonValue": "$.MetricData.AVG_INTERACTION_TIME",
            "Operands": [0],
            "FilterClause": {
                "LogicOperator": "AND",
                "Filters": [{
                    "Type": "MetricDataFilterByQueue",
                    "Data": ["e7a11ddd-041d-449a-ab1e-6a493cb83626", "26b9c7f4-
ed5c-4cd1-b751-2cbfb4c54ea2"]
                }, {
                    "Type": "MetricDataDurationSeconds",
                    "Data": {
                        "Past": 300
                    }
                }, {
                    "Type": "MetricDataFilterByChannel",
                    "Data": ["VOICE"]
                }]
            },
            "Negate": false
        }, {
            "Operator": "NumberGreaterOrEqualTo",
            "ComparisonValue": "$.MetricData.AVG_HOLD_TIME",
            "Operands": [0],
            "FilterClause": {
                "LogicOperator": "AND",
                "Filters": [{
                    "Type": "MetricDataFilterByQueue",
                    "Data": ["e7a11ddd-041d-449a-ab1e-6a493cb83626", "26b9c7f4-
ed5c-4cd1-b751-2cbfb4c54ea2"]
```

```
                }, {
                    "Type": "MetricDataDurationSeconds",
                    "Data": {
                        "Past": 300
                    }
                }, {
                    "Type": "MetricDataFilterByChannel",
                    "Data": ["VOICE"]
                }]
            },
            "Negate": false
        }, {
            "Operator": "NumberGreaterOrEqualTo",
            "ComparisonValue": "$.MetricData.SERVICE_LEVEL",
            "Operands": [0],
            "FilterClause": {
                "LogicOperator": "AND",
                "Filters": [{
                    "Type": "MetricDataFilterByQueue",
                    "Data": ["e7a11ddd-041d-449a-ab1e-6a493cb83626", "26b9c7f4-
ed5c-4cd1-b751-2cbfb4c54ea2"]
                }, {
                    "Type": "MetricDataDurationSeconds",
                    "Data": {
                        "Past": 300
                    }
                }, {
                    "Type": "MetricDataThresholdSeconds",
                    "Data": {
                        "Comparison": "LT",
                        "ThresholdValue": 60
                    }
                }, {
                    "Type": "MetricDataFilterByChannel",
                    "Data": ["VOICE"]
                }]
            },
            "Negate": false
        }]
    }],
    "FilterClause": {
        "LogicOperator": "AND",
        "Filters": [{
            "Type": "MetricDataGrouping",
```

```
            "Data": ["QUEUE"]
        }]
    }
}
```

Example 2

A rule with AGENT_OCCUPANCY in MetricGroup 1 and AGENT_ACTIVITY and MetricGroup 2, with End Dimension = AGENT.

```
{
    "Operator": "OR",
    "Operands": [{
        "Operator": "AND",
        "Operands": [{
            "Operator": "NumberGreaterOrEqualTo",
            "ComparisonValue": "$.MetricData.AGENT_OCCUPANCY",
            "Operands": [0],
            "FilterClause": {
                "LogicOperator": "AND",
                "Filters": [{
                    "Type": "MetricDataFilterByAgent",
                    "Data": ["e7a11ddd-041d-449a-ab1e-6a493cb83626", "26b9c7f4-
ed5c-4cd1-b751-2cbfb4c54ea2"]
                }, {
                    "Type": "MetricDataDurationSeconds",
                    "Data": {
                        "Past": 100
                    }
                }]
            },
            "Negate": false
        }]
    }, {
        "Operator": "AND",
        "Operands": [{
            "Operator": "EQUALS",
            "ComparisonValue": "$.MetricData.AGENT_ACTIVITY",
            "Operands": ["ee8c71b9-49c2-4d00-ab6d-244f2d9a5c58"],
            "FilterClause": {
                "LogicOperator": "AND",
                "Filters": [{
                    "Type": "MetricDataFilterByAgentHierarchyLevelOne",
```

```
                    "Data": ["b99ba702-b84b-4279-abad-5ceda9ff2640"]
                }, {
                    "Type": "MetricDataDurationSeconds",
                    "Data": {
                        "Past":600
                    }
                }]
            },
            "Negate": false
        }]
    }],
    "FilterClause": {
        "LogicOperator": "AND",
        "Filters": [{
            "Type": "MetricDataGrouping",
            "Data": ["AGENT"]
        }]
    }
}
```

Example 3

A rule with FLOWS_STARTED, AVG_FLOW_TIME, MAX_FLOW_TIME in MetricGroup 1 and FLOWS_OUTCOME, PERCENT_FLOWS_OUTCOME and MIN_FLOW_TIME in Metric Group 2, both of which have End Dimension = FLOW.

```
{
  "Operator": "OR",
  "Operands": [
    {
      "Operator": "AND",
      "Operands": [
        {
          "Operator": "NumberGreaterOrEqualTo",
          "ComparisonValue": "$.MetricData.FLOWS_STARTED",
          "Operands": [
            10
          ],
          "FilterClause": {
            "LogicOperator": "AND",
            "Filters": [
              {
                "Type": "MetricDataFilterByFlow",
```

```
            "Data": [
              "z2a11ddd-041d-449a-ab1e-6a493cb83621",
              "y1b9c7f4-ed5c-4cd1-b751-2cbfb4c54ea3"
            ]
          },
          {
            "Type": "MetricDataFilterByChannel",
            "Data": [
              "VOICE"
            ]
          },
          {
            "Type": "MetricDataDurationSeconds",
            "Data": {
              "Past": 300
            }
          }
        ]
      },
      "Negate": false
    },
    {
      "Operator": "NumberGreaterOrEqualTo",
      "ComparisonValue": "$.MetricData.AVG_FLOW_TIME",
      "Operands": [
        600
      ],
      "FilterClause": {
        "LogicOperator": "AND",
        "Filters": [
          {
            "Type": "MetricDataFilterByFlow",
            "Data": [
              "z2a11ddd-041d-449a-ab1e-6a493cb83621",
              "y1b9c7f4-ed5c-4cd1-b751-2cbfb4c54ea3"
            ]
          },
          {
            "Type": "MetricDataFilterByFlowOutcomeType",
            "Data": [
              "DROPPED",
              "TRANSFERRED_TO_QUEUE"
            ]
          },
```

```
            {
              "Type": "MetricDataFilterByChannel",
              "Data": [
                "VOICE"
              ]
            },
            {
              "Type": "MetricDataDurationSeconds",
              "Data": {
                "Past": 300
              }
            }
          ]
        },
        "Negate": false
      },
      {
        "Operator": "NumberGreaterOrEqualTo",
        "ComparisonValue": "$.MetricData.MAX_FLOW_TIME",
        "Operands": [
          600
        ],
        "FilterClause": {
          "LogicOperator": "AND",
          "Filters": [
            {
              "Type": "MetricDataFilterByFlow",
              "Data": [
                "z2a11ddd-041d-449a-ab1e-6a493cb83621",
                "y1b9c7f4-ed5c-4cd1-b751-2cbfb4c54ea3"
              ]
            },
            {
              "Type": "MetricDataFilterByFlowOutcomeType",
              "Data": [
                "DROPPED",
                "TRANSFERRED_TO_QUEUE"
              ]
            },
            {
              "Type": "MetricDataFilterByChannel",
              "Data": [
                "VOICE"
              ]
```

```
            },
            {
              "Type": "MetricDataDurationSeconds",
              "Data": {
                "Past": 300
              }
            }
          ]
        },
        "Negate": false
      }
    ]
  },
  {
    "Operator": "OR",
    "Operands": [
      {
        "Operator": "NumberGreaterOrEqualTo",
        "ComparisonValue": "$.MetricData.FLOWS_OUTCOME",
        "Operands": [
          10
        ],
        "FilterClause": {
          "LogicOperator": "AND",
          "Filters": [
            {
              "Type": "MetricDataFilterByFlow",
              "Data": [
                "z2a11ddd-041d-449a-ab1e-6a493cb83621",
                "y1b9c7f4-ed5c-4cd1-b751-2cbfb4c54ea3"
              ]
            },
            {
              "Type": "MetricDataFilterByFlowOutcomeType",
              "Data": [
                "DROPPED",
                "TRANSFERRED_TO_QUEUE"
              ]
            },
            {
              "Type": "MetricDataFilterByChannel",
              "Data": [
                "VOICE"
              ]
```

```
            },
            {
              "Type": "MetricDataDurationSeconds",
              "Data": {
                "Past": 300
              }
            }
          ]
        },
        "Negate": false
      },
      {
        "Operator": "NumberGreaterOrEqualTo",
        "ComparisonValue": "$.MetricData.PERCENT_FLOWS_OUTCOME",
        "Operands": [
          50
        ],
        "FilterClause": {
          "LogicOperator": "AND",
          "Filters": [
            {
              "Type": "MetricDataFilterByFlow",
              "Data": [
                "z2a11ddd-041d-449a-ab1e-6a493cb83621",
                "y1b9c7f4-ed5c-4cd1-b751-2cbfb4c54ea3"
              ]
            },
            {
              "Type": "MetricDataFilterByFlowOutcomeType",
              "Data": [
                "DROPPED",
                "TRANSFERRED_TO_QUEUE"
              ]
            },
            {
              "Type": "MetricDataFilterByChannel",
              "Data": [
                "VOICE"
              ]
            },
            {
              "Type": "MetricDataDurationSeconds",
              "Data": {
                "Past": 300
```

```
            }
          }
        ]
      },
      "Negate": false
    },
    {
      "Operator": "NumberGreaterOrEqualTo",
      "ComparisonValue": "$.MetricData.MIN_FLOW_TIME",
      "Operands": [
        600
      ],
      "FilterClause": {
        "LogicOperator": "AND",
        "Filters": [
          {
            "Type": "MetricDataFilterByFlow",
            "Data": [
              "z2a11ddd-041d-449a-ab1e-6a493cb83621",
              "y1b9c7f4-ed5c-4cd1-b751-2cbfb4c54ea3"
            ]
          },
          {
            "Type": "MetricDataFilterByFlowOutcomeType",
            "Data": [
              "DROPPED",
              "TRANSFERRED_TO_QUEUE"
            ]
          },
          {
            "Type": "MetricDataFilterByChannel",
            "Data": [
              "VOICE"
            ]
          },
          {
            "Type": "MetricDataDurationSeconds",
            "Data": {
              "Past": 300
            }
          }
        ]
      },
      "Negate": false
```

```
        }
      ]
    }
  ],
  "FilterClause": {
    "LogicOperator": "AND",
    "Filters": [
      {
        "Type": "MetricDataGrouping",
        "Data": [
          "FLOW"
        ]
      }
    ]
  }
}
```

OnContactEvaluationSubmit

Agent Hierarchy

Parameters

- Operator - "CONTAINS_ANY"
- Operands - Agent hierarchy ARNs.
- ComparisonValue - "$.ContactLens.ContactEvaluation.Agent.HierarchyGroup.ARN"
- Negate - false

Initiation Method

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A validated enum set of possible values. Possible values are as follows:
  - 'INBOUND'
  - 'OUTBOUND'
  - 'TRANSFER'
  - 'QUEUE_TRANSFER'

- 'CALLBACK'
- 'API'
- 'DISCONNECT'

- ComparisonValue - "$.ContactLens.ContactEvaluation.InitiationMethod"
- Negate - false

DisconnectReason

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A validated enum set of possible values. Possible values are as follows:
  - 'TELECOM_BUSY'
  - 'TELECOM_NUMBER_INVALID'
  - 'TELECOM_POTENTIAL_BLOCKING'
  - 'TELECOM_UNANSWERED'
  - 'TELECOM_TIMEOUT'
  - 'TELECOM_ORIGINATOR_CANCEL'
  - 'TELECOM_PROBLEM'
  - 'CUSTOMER_NEVER_ARRIVED'
  - 'THIRD_PARTY_DISCONNECT'
  - 'CUSTOMER_DISCONNECT'
  - 'AGENT_DISCONNECT'
  - 'BARGED'
  - 'CONTACT_FLOW_DISCONNECT'
  - 'OTHER'
  - 'OUTBOUND_DESTINATION_ENDPOINT_ERROR'
  - 'OUTBOUND_RESOURCE_ERROR'
  - 'OUTBOUND_ATTEMPT_FAILED'
  - 'EXPIRED'
  - 'AGENT_NETWORK_DISCONNECT'
  - 'CUSTOMER_CONNECTION_NOT_ESTABLISHED'

- 'API'
- 'IDLE_DISCONNECT'
- 'SYSTEM_ERROR'
- 'AGENT_COMPLETED'
- 'TRANSFERRED'
- 'DISCARDED'

- ComparisonValue - "$.ContactLens.ContactEvaluation.DisconnectReason"
- Negate - false

Routing Profile

Parameters

- Operator - "CONTAINS_ANY"
- Operands - Routing profile ARNs.
- ComparisonValue - "$.ContactLens.ContactEvaluation.Agent.RoutingProfile"
- Negate - false

PotentialDisconnectIssue

Parameters

- Operator - "CONTAINS_ANY" or "EQUALS"
- Operands - A validated enum of possible values.
- ComparisonValue - "$.ContactLens.ContactEvaluation.PotentialDisconnectIssue"
- Negate - false

Custom User-Defined Segment Attribute

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A list of segment attribute values. The values must be a value in the pre-defined attribute resource.

- ComparisonValue - "$.ContactLens.ContactEvaluation.SegmentAttributes.UserDefined.[KEY]"

  The KEY must be an instance pre-defined attribute resource.
- Negate - false or true

ContactEvaluation - Results available condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is the evaluation form ID.
- ComparisonValue – "$.ContactLens.ContactEvaluation.Form.FormId"
- Negate - false

ContactEvaluation - Form score condition

ContactEvaluation form score condition has a compound condition format where Operator is an AND condition and its operands consist of two conditions that represent the form and the form score.

Parameters that represent the form: See ContactEvaluation - Results available condition

Parameters that represent the form score:

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is the form score.
- ComparisonValue – "$.ContactLens.ContactEvaluation.Form.Score"
- Negate - false

Following is an example.

```
{
    "Operator": "AND",
    "Operands": [
        {
            "Operator": "EQUALS",
            "Operands": ["11111111-1234-5678-9123-12345678012"],
```

```
            "ComparisonValue": "$.ContactLens.ContactEvaluation.Form.FormId",
            "Negate": false
        },
        {
            "Operator": "NumberLessOrEqualTo",
            "Operands": [50],
            "ComparisonValue": "$.ContactLens.ContactEvaluation.Form.Score",
            "Negate":false
        },
    ]
}
```

ContactEvaluation - Section Score

ContactEvaluation section score condition has a compound condition format where Operator is an AND condition and its operands consist of three conditions that represent the form, section, and section score respectively.

Parameters that represent the form: See ContactEvaluation - Results available condition

Parameters that represent the section:

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is the section reference ID.
- ComparisonValue – "$.ContactLens.ContactEvaluation.Section.SectionRefId"
- Negate - false

Parameters that represent the section score:

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is the section score.
- ComparisonValue – "$.ContactLens.ContactEvaluation.Section.Score"
- Negate - false

Following is an example.

```
{
    "Operator": "AND",
    "Operands": [
```

```
        {
            "Operator": "EQUALS",
            "Operands": ["11111111-1234-5678-9123-12345678012"],
            "ComparisonValue": "$.ContactLens.ContactEvaluation.Form.FormId",
            "Negate": false
        },
        {
            "Operator": "EQUALS",
            "Operands": ["s12345678"],
            "ComparisonValue": "$.ContactLens.ContactEvaluation.Section.SectionRefId",
            "Negate":false
        },
        {
            "Operator": "NumberLessOrEqualTo",
            "Operands": [50],
            "ComparisonValue": "$.ContactLens.ContactEvaluation.Section.Score",
            "Negate":false
        },
    ]
}
```

ContactEvaluation - Question and Answer

ContactEvaluation question and answer condition has a compound condition format where Operator is an AND condition and its operands consist of three conditions that represent the form, question, and answer value respectively.

Parameters that represent the form: See ContactEvaluation - Results available condition

Parameters that represent the question:

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is the question reference ID.
- ComparisonValue – "$.ContactLens.ContactEvaluation.Question.QuestionRefId"
- Negate - false

Parameters that represent a numeric answer:

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is the answer value.

- ComparisonValue – "$.ContactLens.ContactEvaluation.Question.Answer.Value"
- Negate - false

Following is an example for a numeric question type.

```
{
    "Operator": "AND",
    "Operands": [
        {
            "Operator": "EQUALS",
            "Operands": ["11111111-1234-5678-9123-12345678012"],
            "ComparisonValue": "$.ContactLens.ContactEvaluation.Form.FormId",
            "Negate": false
        },
        {
            "Operator": "EQUALS",
            "Operands": ["s12345678"],
            "ComparisonValue":
 "$.ContactLens.ContactEvaluation.Question.QuestionRefId",
            "Negate":false
        },
        {
            "Operator": "NumberLessOrEqualTo",
            "Operands": [5],
            "ComparisonValue": "$.ContactLens.ContactEvaluation.Question.Answer.Value",
            "Negate":false
        }
    ]
}
```

Parameters that represent a single select answer:

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is the answer reference ID.
- ComparisonValue – "$.ContactLens.ContactEvaluation.Question.Answer.ValueRefId"
- Negate - true/false. If set to true, it means *The answer is not equal to the answer reference ID specified in the Operands*.

Following is an example for single select question type.

```
{
    "Operator": "AND",
    "Operands": [
        {
            "Operator": "EQUALS",
            "Operands": ["11111111-1234-5678-9123-12345678012"],
            "ComparisonValue": "$.ContactLens.ContactEvaluation.Form.FormId",
            "Negate": false
        },
        {
            "Operator": "EQUALS",
            "Operands": ["q12345678"],
            "ComparisonValue":
 "$.ContactLens.ContactEvaluation.Question.QuestionRefId",
            "Negate":false
        },
        { // for single select question type
            "Operator": "EQUALS",
            "Operands": ["o12345678"],
            "ComparisonValue":
 "$.ContactLens.ContactEvaluation.Question.Answer.ValueRefId",
            "Negate":false
       },
    ]
}
```

ContactEvaluation - agent condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of agent IDs.
- ComparisonValue – "$.ContactLens.ContactEvaluation.Agent.AgentId"
- Negate - false

ContactEvaluation - queue condition

Parameters

- Operator – "EQUALS"

- Operands – No value
- ComparisonValue – "$.ContactLens.ContactEvaluation.Queue.QueueId"
- Negate - false

ContactEvaluation - contact attributes condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is the contact attribute value.
- ComparisonValue – "$.ContactLens.ContactEvaluation.ContactAttribute.*YOUR_ATTRIBUTE_KEY*"
- Negate - true/false. If set to true, it means *YOUR_ATTRIBUTE_KEY* does not equal to the attribute value specified in the Operands.

OnPostCallAnalysisAvailable

Agent Hierarchy

Parameters

- Operator - "CONTAINS_ANY"
- Operands - Agent hierarchy ARNs.
- ComparisonValue - "$.ContactLens.PostCall.Agent.HierarchyGroup.ARN"
- Negate - false

NonTalkTimePct

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands - An array of numbers, array length can only be 1. Value is an integer between 0 and 100.
- ComparisonValue - "$.ContactLens.PostCall.Agent.NonTalkTimePct"
- Negate - false

Customer Hold Duration Pct

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands - An array of number, array length can only be 1. Value is an integer between 0 and 100.
- ComparisonValue - "$.ContactLens.PostCall.Agent.CustomerHoldDurationPct"
- Negate - false

Initiation Method

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A validated enum set of possible values. Possible values are as follows:
  - 'INBOUND'
  - 'OUTBOUND'
  - 'TRANSFER'
  - 'QUEUE_TRANSFER'
  - 'CALLBACK'
  - 'API'
  - 'DISCONNECT'
- ComparisonValue - "$.ContactLens.PostCall.InitiationMethod"
- Negate - false

DisconnectReason

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A validated enum set of possible values. Possible values are as follows:
  - 'TELECOM_BUSY'
  - 'TELECOM_NUMBER_INVALID'

- 'TELECOM_POTENTIAL_BLOCKING'
- 'TELECOM_UNANSWERED'
- 'TELECOM_TIMEOUT'
- 'TELECOM_ORIGINATOR_CANCEL'
- 'TELECOM_PROBLEM'
- 'CUSTOMER_NEVER_ARRIVED'
- 'THIRD_PARTY_DISCONNECT'
- 'CUSTOMER_DISCONNECT'
- 'AGENT_DISCONNECT'
- 'BARGED'
- 'CONTACT_FLOW_DISCONNECT'
- 'OTHER'
- 'OUTBOUND_DESTINATION_ENDPOINT_ERROR'
- 'OUTBOUND_RESOURCE_ERROR'
- 'OUTBOUND_ATTEMPT_FAILED'
- 'EXPIRED'
- 'AGENT_NETWORK_DISCONNECT'
- 'CUSTOMER_CONNECTION_NOT_ESTABLISHED'
- 'API'
- 'IDLE_DISCONNECT'
- 'SYSTEM_ERROR'
- 'AGENT_COMPLETED'
- 'TRANSFERRED'
- 'DISCARDED'

- ComparisonValue - "$.ContactLens.PostCall.DisconnectReason"
- Negate - false

Absolute Talk Time

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"

- Operands - An integer.
- ComparisonValue - "$.ContactLens.PostCall.TalkTime.TotalTimeSecs"
- FilterClause -

```
{ // required!!
        LogicOperator: 'AND', // has to be 'AND'
        Filters: [ // only allows one of the object
          {
            Type: 'ParticipantRole',
            Data: 'AGENT', // 'AGENT' or 'CUSTOMER'
          },
        ],
      },
```

- Negate - false

Maximum (Highest) Loudness Score

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands - an integer (decibels).
- ComparisonValue - "$.ContactLens.PostCall.Loudness.HighestLoudnessScore"
- FilterClause -

```
{
        LogicOperator: 'AND', // has to be AND
        Filters: [ // only one object
          {
            Type: 'ParticipantRole',
            Data: 'CUSTOMER', // or 'AGENT'
          },
        ],
      },
```

- Negate - false

After Contact Work Duration

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands - A positive integer or zero.
- ComparisonValue - "$.ContactLens.PostCall.Agent.AfterContactWorkDurationSecs"
- Negate - false

Routing profile

Parameters

- Operator - "CONTAINS_ANY"
- Operands - Routing profile ARNs:
- ComparisonValue - "$.ContactLens.PostCall.Agent.RoutingProfile"
- Negate - false

PotentialDisconnectIssue

Parameters

- Operator - "CONTAINS_ANY" | "EQUALS"
- Operands - A validated enum of possible values.
- ComparisonValue - "$.ContactLens.PostCall.PotentialDisconnectIssue"
- Negate - false

Custom User-Defined Segment Attribute

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A list of segment attribute values. The values must be a value in the pre-defined attribute resource.
- ComparisonValue - "$.ContactLens.PostCall.SegmentAttributes.UserDefined.[KEY]"

The KEY must be an instance pre-defined attribute resource.

- Negate - false or true

Connect AI Agent

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A list of strings. Strings are either a list of different UUIDs by themselves or a list of strings prefixed with the same UUID, with different possible version numbers uuid:2 and uuid:3. For example:
  - Different UUIDs by themselves: ['30682191-2312-4c27-9fc1-752e7a399d52', '2b81c7d5-85bc-459f-b605-0f5280a4301e']
  - Prefixed with the same UUID with different possible version numbers: ['30682191-2312-4c27-9fc1-752e7a399d52:1', '30682191-2312-4c27-9fc1-752e7a399d52:2']
- ComparisonValue - "$.ContactLens.PostCall.AiAgent.IdWithVersion"
- FilterClause -

```
[
          { // REQUIRED filter
            Type: 'AiAgentUseCase',
            Data: {
              Value: 'SelfService', // or 'AgentAssistance'
            },
          },
          { // OPTIONAL filter
            Type: 'AiAgentEscalatedToHuman',
            Data: 'TRUE', // or 'FALSE'
          },
        ],
      },
```

- Negate - false

PostCall words or phrases - Exact match

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of strings
- ComparisonValue – "$.ContactLens.PostCall.ExactMatch.Transcript"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "PostCallContactPeriodSeconds", // Optional filter type. If
 omitted, the rule applies to the full contact
            "Data": { // Either "First" or "Last" needs to be specified.
                "First": number
             }
        },
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        }
    ]
}
```

- Negate - true or false. If set to true, it means *If transcript does not contain any of the words mentioned in the Operands*.

PostCall words or phrases - Semantic match

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of strings
- ComparisonValue – "$.ContactLens.PostCall.SemanticMatch.Transcript"
- FilterClause –

```
{
```

```
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "PostCallContactPeriodSeconds", // Optional filter type. If
 omitted, the rule applies to the full contact
            "Data": { // Either "First" or "Last" needs to be specified.
                "First": number
             }
        },
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        }
    ]
}
```

- Negate - true or false. If set to true, it means *If transcript does not contain any of the words mentioned in the Operands*.

PostCall natural language - Semantic match

Parameters

- Operator - "EQUALS"
- Operands – A list with a single string
- ComparisonValue – "$.ContactLens.PostCall.SemanticMatch.Phrase"
- FilterClause –

```
{
     Operator: "EQUALS",
     Operands: [
          "Customer called to change their address and agent proceeded with the
 address change during the call."
     ],
     ComparisonValue: "$.ContactLens.PostCall.SemanticMatch.Phrase",
     Negate: true or false
},
```

- Negate - true or false. If set to true, it means *If transcript does not contain any of the words mentioned in the Operands*.

PostCall words or phrases - Pattern match

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of pattern match objects. See PatternMatch Operands.
- ComparisonValue – "$.ContactLens.PostCall.PatternMatch.Transcript"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "PostCallContactPeriodSeconds", // Optional filter type. If
 omitted, the rule applies to the full contact
            "Data": { // Either "First" or "Last" needs to be specified.
                "First": number
             }
        },
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
        {
            "Type": "PatternMatchLanguageFilter",
            "Data": "EN"|"ES"|"AR"|"DE"|"FR"|"HI"|"IT"|"PT"|"KO"|"JA"|"ZH"
        }
    ]
}
```

- Negate - true or false. If set to true, it means *If transcript does not contain any of the words mentioned in the Operands*.

PostCall agent condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of agent IDs
- ComparisonValue – "$.ContactLens.PostCall.Agent.AgentId"

- Negate - false

PostCall queue condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of queue IDs.
- ComparisonValue – "$.ContactLens.PostCall.Queue.QueueId"
- Negate - false

PostCall no queue condition

Parameters

- Operator - "EQUALS"
- Operands – No value.
- ComparisonValue – "$.ContactLens.PostCall.Queue.QueueId"
- Negate - false

PostCall contact attributes condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is the contact attribute value.
- ComparisonValue – "$.ContactLens.PostCall.ContactAttribute.*YOUR_ATTRIBUTE_KEY*"
- Negate - true/false. If set to true, it means *YOUR_ATTRIBUTE_KEY does not equal to the attribute value specified in the Operands.* .

PostCall sentiment state condition

Parameters

- Operator - "EQUALS"

- Operands – An array of string, array length can only be 1. Value is one of "POSITIVE", "NEGATIVE", "NEUTRAL".
- ComparisonValue – "$.ContactLens.PostCall.Sentiment.State"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
    ]
}
```

- Negate - false

PostCall sentiment overall score condition

Parameters

- Operator - "NumberLessOrEqualTo | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer between -5 and 5, inclusive.
- ComparisonValue – "$.ContactLens.PostCall.Sentiment.OverallScore"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
    ]
}
```

- Negate - false

PostCall sentiment score shift condition

PostCall sentiment score shift condition has a compound condition format where Operator is an AND condition and its operands consist of two conditions that represent beginning and end score respectively.

Parameters that represent the beginning score

- Operator - "NumberLessOrEqualTo | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer between -5 and 5, inclusive.
- ComparisonValue – "$.ContactLens.PostCall.Sentiment.Score.Beginning"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
    ]
}
```

- Negate - false

Parameters that represent the end score

- Operator - "NumberLessOrEqualTo | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer between -5 and 5, inclusive.
- ComparisonValue – "$.ContactLens.PostCall.Sentiment.Score.End"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "ParticipantRole",
```

```
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
    ]
}
```

- Negate - false

Following is an example:

```
{
        "Operator": "AND",
        "Operands": [
          {
            "Operator": "NumberGreaterOrEqualTo",
            "Operands": [
              2
            ],
            "ComparisonValue": "$.ContactLens.PostCall.Sentiment.Score.Beginning",
            "FilterClause": {
              "LogicOperator": "AND",
              "Filters": [
                {
                  "Type": "ParticipantRole",
                  "Data": "AGENT"
                }
              ]
            },
            "Negate": false
          },
          {
            "Operator": "NumberGreaterOrEqualTo",
            "Operands": [
              3
            ],
            "ComparisonValue": "$.ContactLens.PostCall.Sentiment.Score.End",
            "FilterClause": {
              "LogicOperator": "AND",
              "Filters": [
                {
                  "Type": "ParticipantRole",
                  "Data": "AGENT"
                }
              ]
```

```
            },
            "Negate": false
          }
        ]
      }
```

Parameters for overall score

- Operator - "NumberLessOrEqualTo | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer between -5 and 5, inclusive.
- ComparisonValue – "$.ContactLens.PostCall.Sentiment.OverallScore"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
    ]
}
```

- Negate - false

PostCall total non-talk time condition

Parameters

- Operator - "NumberLessOrEqualTo | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.PostCall.NonTalkTime.TotalTimeSecs"
- Negate - false

PostCall longest non-talk time condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.PostCall.NonTalkTime.LongestTimeSecs"
- Negate - false

PostCall interruption condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.PostCall.Interruptions.Instances"
- Negate - false

PostCall total hold time condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.PostCall.Agent.CustomerHoldDurationSecs"
- Negate - false

PostCall longest hold time condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"

- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.PostCall.Agent.LongestHoldDurationSecs"
- Negate - false

PostCall number of holds condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.PostCall.Agent.NumberOfHolds"
- Negate - false

PostCall agent interaction duration condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.PostCall.Agent.AgentInteractionDurationSecs"
- Negate - false

OnRealTimeCallAnalysisAvailable

RealTimeCall words or phrases - Exact match

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of strings
- ComparisonValue – "$.ContactLens.RealTimeCall.ExactMatch.Transcript"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        }
    ]
}
```

- Negate - false.

RealTimeCall words or phrases - Pattern match

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of pattern match objects. See PatternMatch Operands.
- ComparisonValue – "$.ContactLens.RealTimeCall.PatternMatch.Transcript"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
        {
            "Type": "PatternMatchLanguageFilter",
            "Data": "EN"|"ES"|"AR"|"DE"|"FR"|"HI"|"IT"|"PT"|"KO"|"JA"|"ZH"
        }
    ]
}
```

- Negate - false

RealTimeCall agent condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of agent IDs
- ComparisonValue – "$.ContactLens.RealTimeCall.Agent.AgentId"
- Negate - false

RealTimeCall queue condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of queue IDs
- ComparisonValue – "$.ContactLens.RealTimeCall.Queue.QueueId"
- Negate - true/false. If set to true, it means *If queue is not any of the queues mentioned in the Operands*.

RealTimeCall contact attributes condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is the contact attribute value.
- ComparisonValue – "$.ContactLens.RealTimeCall.ContactAttribute.*YOUR_ATTRIBUTE_KEY*"
- Negate - true/false. If set to true, it means *YOUR_ATTRIBUTE_KEY does not equal to the attribute value specified in the Operands* .

RealTimeCall sentiment state condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is one of "POSITIVE", "NEGATIVE", "NEUTRAL".

- ComparisonValue – "$.ContactLens.RealTimeCall.Sentiment.State"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
        {
            "Type": "RealTimeCallContactPeriodSeconds",
            "Data": {
                "Past": number
            }
        }
    ]
}
```

- Negate - false

OnPostChatAnalysisAvailable

Agent Hierarchy

Parameters

- Operator - "CONTAINS_ANY"
- Operands - Agent hierarchy ARNs.
- ComparisonValue - "$.ContactLens.PostChat.Agent.HierarchyGroup.ARN"
- Negate - false

Initiation Method

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A validated enum set of possible values. Possible values are as follows:

- 'INBOUND'
- 'OUTBOUND'
- 'TRANSFER'
- 'QUEUE_TRANSFER'
- 'CALLBACK'
- 'API'
- 'DISCONNECT'

- ComparisonValue - "$.ContactLens.PostChat.InitiationMethod"
- Negate - false

DisconnectReason

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A validated enum set of possible values. Possible values are as follows:
  - 'TELECOM_BUSY'
  - 'TELECOM_NUMBER_INVALID'
  - 'TELECOM_POTENTIAL_BLOCKING'
  - 'TELECOM_UNANSWERED'
  - 'TELECOM_TIMEOUT'
  - 'TELECOM_ORIGINATOR_CANCEL'
  - 'TELECOM_PROBLEM'
  - 'CUSTOMER_NEVER_ARRIVED'
  - 'THIRD_PARTY_DISCONNECT'
  - 'CUSTOMER_DISCONNECT'
  - 'AGENT_DISCONNECT'
  - 'BARGED'
  - 'CONTACT_FLOW_DISCONNECT'
  - 'OTHER'
  - 'OUTBOUND_DESTINATION_ENDPOINT_ERROR'
  - 'OUTBOUND_RESOURCE_ERROR'

- 'OUTBOUND_ATTEMPT_FAILED'
- 'EXPIRED'
- 'AGENT_NETWORK_DISCONNECT'
- 'CUSTOMER_CONNECTION_NOT_ESTABLISHED'
- 'API'
- 'IDLE_DISCONNECT'
- 'SYSTEM_ERROR'
- 'AGENT_COMPLETED'
- 'TRANSFERRED'
- 'DISCARDED'

- ComparisonValue - "$.ContactLens.PostChat.DisconnectReason"
- Negate - false

After Contact Work Duration

Parameters

- Operator - "NumberGreaterOrEqualTo" | "NumberLessOrEqualTo"
- Operands - A positive integer or zero.
- ComparisonValue - "$.ContactLens.PostChat.Agent.AfterContactWorkDurationSecs"
- Negate - false

Routing Profile

Parameters

- Operator - "CONTAINS_ANY"
- Operands - Routing profile ARNs.
- ComparisonValue - "$.ContactLens.PostChat.Agent.RoutingProfile"
- Negate - false

PotentialDisconnectIssue

Parameters

- Operator - "CONTAINS_ANY" where it is matched by a possible operand or "EQUALS" where the operand has to have no value.
- Operands - A validated enum of possible values or no value.
- ComparisonValue - "$.ContactLens.PostChat.PotentialDisconnectIssue"
- Negate - false

Custom User-Defined Segment Attribute

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A list of segment attribute values. The values must be a value in the pre-defined attribute resource.
- ComparisonValue - "$.ContactLens.PostChat.SegmentAttributes.UserDefined.[KEY]"

  The KEY must be an instance pre-defined attribute resource.
- Negate - false

Connect AI Agent

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A list of strings. Strings are either a list of different UUIDs by themselves or a list of strings prefixed with the same UUID, with different possible version numbers uuid:2 and uuid:3. For example:
  - Different UUIDs by themselves: ['30682191-2312-4c27-9fc1-752e7a399d52', '2b81c7d5-85bc-459f-b605-0f5280a4301e']
  - Prefixed with the same UUID with different possible version numbers: ['30682191-2312-4c27-9fc1-752e7a399d52:1', '30682191-2312-4c27-9fc1-752e7a399d52:2']
- ComparisonValue - "$.ContactLens.PostChat.AiAgent.IdWithVersion"

- FilterClause -

```
{
        LogicOperator: 'AND', // has to be 'AND'
        Filters: [
          { // REQUIRED filter
            Type: 'AiAgentUseCase',
            Data: {
              Value: 'SelfService', // or 'AgentAssistance'
            },
          },
          { // OPTIONAL filter
            Type: 'AiAgentEscalatedToHuman',
            Data: 'TRUE', // or 'FALSE'
          },
        ],
      },
```

- Negate - false

PostChat words or phrases - Exact match

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of strings
- ComparisonValue – "$.ContactLens.PostChat.ExactMatch.Transcript"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "PostChatContactPeriodSeconds", // Optional filter type. If
 omitted, the rule applies to the full contact
            "Data": { // Either "First" or "Last" needs to be specified.
                "First": number
             }
        },
        {
```

```
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        }
    ]
}
```

- Negate - true or false. If set to true, it means *If transcript does not contain any of the words mentioned in the Operands*.

PostChat words or phrases - Semantic match

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of strings
- ComparisonValue – "$.ContactLens.PostChat.SemanticMatch.Transcript"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "PostChatContactPeriodSeconds", // Optional filter type. If
 omitted, the rule applies to the full contact
            "Data": { // Either "First" or "Last" needs to be specified.
                "First": number
             }
        },
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        }
    ]
}
```

- Negate - true or false. If set to true, it means *If transcript does not contain any of the words mentioned in the Operands*.

PostChat natural language - Semantic match

Parameters

- Operator - "EQUALS"
- Operands – A list with a single string
- ComparisonValue – "$.ContactLens.PostChat.SemanticMatch.Phrase"
- FilterClause –

```
{
     Operator: "EQUALS",
     Operands: [
          "Customer called to change their address and agent proceeded with the
 address change during the call."
     ],
     ComparisonValue: "$.ContactLens.PostChat.SemanticMatch.Phrase",
     Negate: true or false
},
```

- Negate - true or false. If set to true, it means *If transcript does not contain any of the words mentioned in the Operands*.

PostChat words or phrases - Pattern match

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of pattern match objects. See PatternMatch Operands.
- ComparisonValue – "$.ContactLens.PostChat.PatternMatch.Transcript"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "PostChatContactPeriodSeconds", // Optional filter type. If
 omitted, the rule applies to the full contact
            "Data": { // Either "First" or "Last" needs to be specified.
                "First": number
             }
```

```
        },
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
        {
            "Type": "PatternMatchLanguageFilter",
            "Data": "EN"|"ES"|"AR"|"DE"|"FR"|"HI"|"IT"|"PT"|"KO"|"JA"|"ZH"
        }
    ]
}
```

- Negate - true or false. If set to true, it means *If the transcript does not contain any of the words mentioned in the Operands*.

PostChat agent condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of agent IDs
- ComparisonValue – "$.ContactLens.PostChat.Agent.AgentId"
- Negate - false

PostChat queue condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of queue IDs.
- ComparisonValue – "$.ContactLens.PostChat.Queue.QueueId"
- Negate - false

PostChat no queue condition

Parameters

- Operator - "EQUALS"

- Operands – No value.
- ComparisonValue – "$.ContactLens.PostChat.Queue.QueueId"
- Negate - false

PostChat contact attributes condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is the contact attribute value.
- ComparisonValue – "$.ContactLens.PostChat.ContactAttribute.*YOUR_ATTRIBUTE_KEY*"
- Negate - true/false. If set to true, it means *YOUR_ATTRIBUTE_KEY does not equal to the attribute value specified in the Operands* .

PostChat sentiment state condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is one of "POSITIVE", "NEGATIVE", "NEUTRAL".
- ComparisonValue – "$.ContactLens.PostChat.Sentiment.State"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
    ]
}
```

- Negate - false

PostChat sentiment overall score condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer between -5 and 5, inclusive.
- ComparisonValue – "$.ContactLens.PostChat.Sentiment.OverallScore"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
    ]
}
```

- Negate - false

Postchat sentiment score shift condition

PostChat sentiment score shift condition has a compound condition format where Operator is an AND condition and its operands consist of two conditions that represent beginning and end score respectively.

Parameters that represent the beginning score

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer between -5 and 5, inclusive.
- ComparisonValue – "$.ContactLens.PostChat.Sentiment.Score.Beginning"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
```

```
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
    ]
}
```

- Negate - false

Parameters that represent the end score

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer between -5 and 5, inclusive.
- ComparisonValue – "$.ContactLens.PostChat.Sentiment.Score.End"
- FilterClause –

```
{
    "LogicOperator": "AND", // Only "AND" is supported
    "Filters": [
        {
            "Type": "ParticipantRole",
            "Data": "CUSTOMER" | "AGENT" | "ANY"
        },
    ]
}
```

- Negate - false

Following is an example:

```
{
        "Operator": "AND",
        "Operands": [
          {
            "Operator": "NumberGreaterOrEqualTo",
            "Operands": [
              2
            ],
            "ComparisonValue": "$.ContactLens.PostChat.Sentiment.Score.Beginning",
```

```
            "FilterClause": {
              "LogicOperator": "AND",
              "Filters": [
                {
                  "Type": "ParticipantRole",
                  "Data": "AGENT"
                }
              ]
            },
            "Negate": false
          },
          {
            "Operator": "NumberGreaterOrEqualTo",
            "Operands": [
              3
            ],
            "ComparisonValue": "$.ContactLens.PostChat.Sentiment.Score.End",
            "FilterClause": {
              "LogicOperator": "AND",
              "Filters": [
                {
                  "Type": "ParticipantRole",
                  "Data": "AGENT"
                }
              ]
            },
            "Negate": false
          }
        ]
      }
```

PostChat agent first response time condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.PostChat.ResponseTime.Agent.FirstResponseTimeSecs"
- Negate - false

PostChat agent average response time condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.PostChat.ResponseTime.Agent.AverageTimeSecs"
- Negate - false

PostChat agent longest response time condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.PostChat.ResponseTime.Agent.LongestTimeSecs"
- Negate - false

PostChat agent interaction duration condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.PostChat.Agent.AgentInteractionDurationSecs"
- Negate - false

OnEmailAnalysisAvailable

Agent Hierarchy

Parameters

- Operator - "CONTAINS_ANY"
- Operands - Agent hierarchy ARNs.
- ComparisonValue - "$.ContactLens.Email.Agent.HierarchyGroup.ARN"
- Negate - false

Initiation Method

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A validated enum set of possible values. Possible values are as follows:
  - 'INBOUND'
  - 'OUTBOUND'
  - 'AGENT_REPLY'
  - 'FLOW'
- ComparisonValue - "$.ContactLens.Email.InitiationMethod"
- Negate - false

DisconnectReason

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A validated enum set of possible values. Possible values are as follows:
  - 'AGENT_DISCONNECT'
  - 'CONTACT_FLOW_DISCONNECT'
  - 'EXPIRED'
  - 'DISCARDED'

- 'OTHER'
- 'API'
- 'TRANSFERRED'

- ComparisonValue - "$.ContactLens.Email.DisconnectReason"
- Negate - false

After Contact Work Duration

Parameters

- Operator - "NumberGreaterOrEqualTo" | "NumberLessOrEqualTo"
- Operands - A positive integer or zero.
- ComparisonValue - "$.ContactLens.Email.Agent.AfterContactWorkDurationSecs"
- Negate - false

Routing Profile

Parameters

- Operator - "CONTAINS_ANY"
- Operands - Routing profile ARNs.
- ComparisonValue - "$.ContactLens.Email.Agent.RoutingProfile"
- Negate - false

Custom User-Defined Segment Attribute

Parameters

- Operator - "CONTAINS_ANY"
- Operands - A list of segment attribute values. The values must be a value in the pre-defined attribute resource.
- ComparisonValue - "$.ContactLens.Email.SegmentAttributes.UserDefined.[KEY]"

  The KEY must be an instance pre-defined attribute resource.
- Negate - false or true

Email agent condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of agent IDs
- ComparisonValue – "$.ContactLens.Email.Agent.AgentId"
- Negate - false

Email queue condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of queue IDs.
- ComparisonValue – "$.ContactLens.Email.Queue.QueueId"
- Negate - false

Email no queue condition

Parameters

- Operator - "EQUALS"
- Operands – No value.
- ComparisonValue – "$.ContactLens.Email.Queue.QueueId"
- Negate - false

Email contact attributes condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is the contact attribute value.
- ComparisonValue – "$.ContactLens.Email.ContactAttribute.*YOUR_ATTRIBUTE_KEY*"

- Negate - true/false. If set to true, it means *`YOUR_ATTRIBUTE_KEY` does not equal to the attribute value specified in the Operands* .

Email agent interaction duration condition

Parameters

- Operator - "NumberLessOrEqualTo" | "NumberGreaterOrEqualTo"
- Operands – An array of number, array length can only be 1. Value is an integer with minimum value 1.
- ComparisonValue – "$.ContactLens.Email.Agent.AgentInteractionDurationSecs"
- Negate - false

OnZendeskTicketCreate

Type condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is one of "question", "incident", "problem", "task"
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.Type"
- Negate - true or false. If set to true, it means *If ticket type does not equal to the type specified in the Operands*.

Priority condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is one of "low", "normal", "high", "urgent"
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.Priority"
- Negate - true or false. If set to true, it means *If ticket priority does not equal to the priority specified in the Operands*.

Status condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is one of "open", "pending", "solved", "close", "hold"
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.Status"
- Negate - true or false. If set to true, it means *If ticket status does not equal to the status specified in the Operands.*.

RequesterId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a requester id.
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.RequesterId"
- Negate - true or false. If set to true, it means *If requester id does not equal to the requester id specified in the Operands.*.

SubmitterId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a submitter id.
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.SubmitterId"
- Negate - true or false. If set to true, it means *If submitter id does not equal to the submitter id specified in the Operands.*.

AssigneeId condition

Parameters

- Operator - "EQUALS"

- Operands – An array of string, array length can only be 1. Value is an assignee id.
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.AssigneeId"
- Negate - true or false. If set to true, it means *If assignee id does not equal to the assignee id specified in the Operands.*.

OrganizationId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is an organization id.
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.OrganizationId"
- Negate - true or false. If set to true, it means *If organization id does not equal to the organization id specified in the Operands.*.

BrandId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a brand id.
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.BrandId"
- Negate - true or false. If set to true, it means *If brand id does not equal to the brand id specified in the Operands*.

FormId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a form id.
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.FormId"
- Negate - true or false. If set to true, it means *If form id does not equal to the form id specified in the Operands*.

ExternalId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is an external id.
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.ExternalId"
- Negate - true or false. If set to true, it means *If external id does not equal to the external id specified in the Operands*.

Channel condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a channel string.
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.Channel"
- Negate - true or false. If set to true, it means *If channel value does not equal to the channel string specified in the Operands*.

Tags condition

Parameters

- Operator - "CONTAINS"
- Operands – An array of string, array length can only be 1. Value is a tag string.
- ComparisonValue – "$.ThirdParty.Zendesk.TicketCreate.Tags"
- Negate - true or false. If set to true, it means *If tag value does not equal to the tag string specified in the Operands*.

OnZendeskTicketStatusUpdate

Type condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is one of "question", "incident", "problem", "task"
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.Type"
- Negate - true or false. If set to true, it means *If ticket type does not equal to the type specified in the Operands*.

Priority condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is one of "low", "normal", "high", "urgent"
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.Priority"
- Negate - true or false. If set to true, it means *If ticket priority does not equal to the priority specified in the Operands*.

Status condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is one of "open", "pending", "solved", "close", "hold"
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.Status"
- Negate - true or false. If set to true, it means *If ticket status does not equal to the status specified in the Operands.*.

RequesterId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a requester id.
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.RequesterId"
- Negate - true or false. If set to true, it means *If requester id does not equal to the requester id specified in the Operands.*.

SubmitterId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a submitter id.
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.SubmitterId"
- Negate - true or false. If set to true, it means *If submitter id does not equal to the submitter id specified in the Operands.*.

AssigneeId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is an assignee id.
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.AssigneeId"
- Negate - true or false. If set to true, it means *If assignee id does not equal to the assignee id specified in the Operands.*.

OrganizationId condition

Parameters

- Operator - "EQUALS"

- Operands – An array of string, array length can only be 1. Value is an organization id.
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.OrganizationId"
- Negate - true or false. If set to true, it means *If organization id does not equal to the organization id specified in the Operands.*.

BrandId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a brand id.
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.BrandId"
- Negate - true or false. If set to true, it means *If brand id does not equal to the brand id specified in the Operands*.

FormId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a form id.
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.FormId"
- Negate - true or false. If set to true, it means *If form id does not equal to the form id specified in the Operands*.

ExternalId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is an external id.
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.ExternalId"
- Negate - true or false. If set to true, it means *If external id does not equal to the external id specified in the Operands*.

Channel condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a channel string.
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.Channel"
- Negate - true or false. If set to true, it means *If channel value does not equal to the channel string specified in the Operands*.

Tags condition

Parameters

- Operator - "CONTAINS"
- Operands – An array of string, array length can only be 1. Value is a tag string.
- ComparisonValue – "$.ThirdParty.Zendesk.StatusUpdate.Tags"
- Negate - true or false. If set to true, it means *If tag value does not equal to the tag string specified in the Operands*.

OnSalesforceCaseCreate

AccountId condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is an account Id string.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.AccountId"
- Negate - true or false. If set to true, it means *If account id does not equal to the account id specified in the Operands*.

Name condition

Parameters

- Operator - "EQUALS" or "CONTAINS"
- Operands – An array of string, array length can only be 1. Value is an account Id string.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.Name"
- Negate - true or false. If set to true, it means *If name does not equal to or does not contain the priority specified in the Operands*.

Email condition

Parameters

- Operator - "EQUALS" or "CONTAINS"
- Operands – An array of string, array length can only be 1. Value is an email string.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.Email"
- Negate - true or false. If set to true, it means *If email does not equal to or does not contain the email specified in the Operands*.

Phone condition

Parameters

- Operator - "EQUALS" or "CONTAINS"
- Operands – An array of string, array length can only be 1. Value is a phone string.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.Phone"
- Negate - true or false. If set to true, it means *If phone does not equal to or does not contain the phone specified in the Operands*.

Company condition

Parameters

- Operator - "EQUALS" or "CONTAINS"

- Operands – An array of string, array length can only be 1. Value is a company string.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.Company"
- Negate - true or false. If set to true, it means *If company does not equal to or does not contain the company specified in the Operands*.

Type condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a type string.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.Type"
- Negate - true or false. If set to true, it means *If type does not equal to the type specified in the Operands*.

Reason condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a reason string.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.Reason"
- Negate - true or false. If set to true, it means *If reason does not equal to the reason specified in the Operands*.

Origin condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is an origin string.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.Origin"
- Negate - true or false. If set to true, it means *If origin does not equal to the origin specified in the Operands*.

Subject condition

Parameters

- Operator - "EQUALS" or "CONTAINS"
- Operands – An array of string, array length can only be 1. Value is a subject string.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.Subject"
- Negate - true or false. If set to true, it means *If subject does not equal to or does not contain the subject specified in the Operands*.

Priority condition

Parameters

- Operator - "EQUALS"
- Operands – An array of string, array length can only be 1. Value is a priority string.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.Priority"
- Negate - true or false. If set to true, it means *If priority does not equal to the priority specified in the Operands*.

Description condition

Parameters

- Operator - "EQUALS" or "CONTAINS"
- Operands – An array of string, array length can only be 1. Value is a description string.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.Description"
- Negate - true or false. If set to true, it means *If description does not equal to or does not contain the description specified in the Operands*.

CustomAttribute condition

Parameters

- Operator - "EQUALS"

- Operands – An array of string, array length can only be 1. Value is the custom attribute value.
- ComparisonValue – "$.ThirdParty.Salesforce.CaseCreate.CustomAttribute.*YOUR_ATTRIBUTE_KEY*"
- Negate - true or false. If set to true, it means *If custom attribute does not equal to the custom attribute specified in the Operands*.

OnCaseCreate

Assigned queue condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of queue ids
- ComparisonValue – "$.Case.Fields.assigned_queue"
- Negate - false

```
{
    "Operator": "CONTAINS_ANY",
    "Operands": ["queueId"],
    "ComparisonValue": "$.Case.Fields.assigned_queue",
    "Negate": false
}
```

Assigned user condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of queue ids
- ComparisonValue – "$.Case.Fields.assigned_user"
- Negate - false

```
{
```

```
    "Operator": "CONTAINS_ANY",
    "Operands": ["userId1, userId2"],
    "ComparisonValue": "$.Case.Fields.assigned_user",
    "Negate": false
}
```

Case reason condition

Parameters

- Operator - "EQUALS"
- Operands – An array of strings in which the array length can only be 1. The value is a case reason.
- ComparisonValue – "$.Case.Fields.case_reason"
- Negate - true/false

```
{
    "Operator": "EQUALS",
    "Operands": ["Refund"],
    "ComparisonValue": "$.Case.Fields.case_reason",
    "Negate": false
}
```

Case status condition

Parameters

- Operator - "EQUALS"
- Operands – An array of strings in which the array length can only be 1. The value is a case status.
- ComparisonValue – "$.Case.Fields.status"
- Negate - false

```
{
    "Operator": "EQUALS",
    "Operands": ["open"],
    "ComparisonValue": "$.Case.Fields.status",
```

```
    "Negate": false
}
```

Case summary condition

Parameters

- Operator - “CONTAINS” | “EQUALS”
- Operands – An array of strings in which the array length can only be 1. The value is a case summary.
- ComparisonValue – "$.Case.Fields.summary"
- Negate - false

```
{
    "Operator": "EQUALS",
    "Operands": ["open"],
    "ComparisonValue": "$.Case.Fields.summary",
    "Negate": false
}
```

Case date/time last closed condition

Parameters

- Operator - “NumberLessOrEqualTo” | “NumberGreaterOrEqualTo”
- Operands – An array of strings in which the array length can only be 1. The value is interpreted as a numeric value.
- ComparisonValue – "$.Case.Fields.last_closed_datetime"
- Negate - false

```
{
    "Operator": "NumberLessOrEqualTo",
    "Operands": ["2023-11-11T11:11:11.111111Z"],
    "ComparisonValue": "$.Case.Fields.last_closed_datetime",
    "Negate": false
```

```
}
```

Case date/time updated condition

Parameters

- Operator - “NumberLessOrEqualTo” | “NumberGreaterOrEqualTo”
- Operands – An array of strings in which the array length can only be 1. The value is interpreted as a numeric value.
- ComparisonValue – "$.Case.Fields.last_updated_datetime"
- Negate - false

```
{
    "Operator": "NumberLessOrEqualTo",
    "Operands": ["2023-11-11T11:11:11.111111Z"],
    "ComparisonValue": "$.Case.Fields.last_updated_datetime",
    "Negate": false
}
```

Case date/time opened condition

Parameters

- Operator - “NumberLessOrEqualTo” | “NumberGreaterOrEqualTo”
- Operands – An array of strings in which the array length can only be 1. The value is interpreted as a numeric value.
- ComparisonValue – "$.Case.Fields.created_datetime"
- Negate - true/false

```
{
    "Operator": "NumberLessOrEqualTo",
    "Operands": ["2023-11-11T11:11:11.111111Z"],
    "ComparisonValue": "$.Case.Fields.created_datetime",
    "Negate": false
}
```

Case reference number condition

Parameters

- Operator - “CONTAINS” | “EQUALS”
- Operands – An array of strings in which the array length can only be 1. The value is interpreted as a numeric value.
- ComparisonValue – "$.Case.Fields.reference_number"
- Negate - true

```
{
    "Operator": "CONTAINS",
    "Operands": ["11111111"],
    "ComparisonValue": "$.Case.Fields.reference_number",
    "Negate": false
}
```

Case customer fields condition

Parameters

- Operator - “CONTAINS” | “EQUALS”
- Operands – An array of strings in which the array length can only be 1.
- ComparisonValue – "$.Case.Fields.custom_case_field_id"
- Negate - true/false

```
{
    "Operator": "EQUALS",
    "Operands": ["vip"],
    "ComparisonValue": "$.Case.Fields.custom_case_field_id",
    "Negate": false
}
```

Case template condition

Parameters

- Operator - “CONTAINS_ANY”
- Operands – A list of template ids.
- ComparisonValue – "$.Case.TemplateId"
- Negate - true/false

```
{
"Operator": "CONTAINS_ANY",
"Operands": ["templateId"],
"ComparisonValue": "$.Case.TemplateId",
"Negate": false
}
```

OnCaseUpdate

Assigned queue condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of queue ids
- ComparisonValue – "$.Case.Fields.assigned_queue"
- Negate - false

```
{
    "Operator": "CONTAINS_ANY",
    "Operands": ["queueId"],
    "ComparisonValue": "$.Case.Fields.assigned_queue",
    "Negate": false
}
```

Assigned user condition

Parameters

- Operator - "CONTAINS_ANY"
- Operands – A list of queue ids
- ComparisonValue – "$.Case.Fields.assigned_user"
- Negate - false

```
{
    "Operator": "CONTAINS_ANY",
    "Operands": ["userId1, userId2"],
    "ComparisonValue": "$.Case.Fields.assigned_user",
    "Negate": false
}
```

Case reason condition

Parameters

- Operator - "EQUALS"
- Operands – An array of strings in which the array length can only be 1. The value is a case reason.
- ComparisonValue – "$.Case.Fields.case_reason"
- Negate - true/false

```
{
    "Operator": "EQUALS",
    "Operands": ["Refund"],
    "ComparisonValue": "$.Case.Fields.case_reason",
    "Negate": false
}
```

Case status condition

Parameters

- Operator - "EQUALS"
- Operands – An array of strings in which the array length can only be 1. The value is a case status.
- ComparisonValue – "$.Case.Fields.status"
- Negate - false

```
{
    "Operator": "EQUALS",
    "Operands": ["open"],
    "ComparisonValue": "$.Case.Fields.status",
    "Negate": false
}
```

Case summary condition

Parameters

- Operator - “CONTAINS” | “EQUALS”
- Operands – An array of strings in which the array length can only be 1. The value is a case summary.
- ComparisonValue – "$.Case.Fields.summary"
- Negate - false

```
{
    "Operator": "EQUALS",
    "Operands": ["open"],
    "ComparisonValue": "$.Case.Fields.summary",
    "Negate": false
}
```

Case date/time last closed condition

Parameters

- Operator - “NumberLessOrEqualTo” | “NumberGreaterOrEqualTo”
- Operands – An array of strings in which the array length can only be 1. The value is interpreted as a numeric value.
- ComparisonValue – "$.Case.Fields.last_closed_datetime"
- Negate - false

```
{
    "Operator": "NumberLessOrEqualTo",
    "Operands": ["2023-11-11T11:11:11.111111Z"],
    "ComparisonValue": "$.Case.Fields.last_closed_datetime",
    "Negate": false
}
```

Case date/time updated condition

Parameters

- Operator - “NumberLessOrEqualTo” | “NumberGreaterOrEqualTo”
- Operands – An array of strings in which the array length can only be 1. The value is interpreted as a numeric value.
- ComparisonValue – "$.Case.Fields.last_updated_datetime"
- Negate - false

```
{
    "Operator": "NumberLessOrEqualTo",
    "Operands": ["2023-11-11T11:11:11.111111Z"],
    "ComparisonValue": "$.Case.Fields.last_updated_datetime",
    "Negate": false
}
```

Case date/time opened condition

Parameters

- Operator - “NumberLessOrEqualTo” | “NumberGreaterOrEqualTo”
- Operands – An array of strings in which the array length can only be 1. The value is interpreted as a numeric value.
- ComparisonValue – "$.Case.Fields.created_datetime"
- Negate - true/false

```
{
    "Operator": "NumberLessOrEqualTo",
    "Operands": ["2023-11-11T11:11:11.111111Z"],
    "ComparisonValue": "$.Case.Fields.created_datetime",
    "Negate": false
}
```

Case reference number condition

Parameters

- Operator - “CONTAINS” | “EQUALS”
- Operands – An array of strings in which the array length can only be 1. The value is interpreted as a numeric value.
- ComparisonValue – "$.Case.Fields.reference_number"
- Negate - true

```
{
    "Operator": "CONTAINS",
    "Operands": ["11111111"],
    "ComparisonValue": "$.Case.Fields.reference_number",
    "Negate": false
}
```

Case customer fields condition

Parameters

- Operator - “CONTAINS” | “EQUALS”
- Operands – An array of strings in which the array length can only be 1.
- ComparisonValue – "$.Case.Fields.custom_case_field_id"
- Negate - true/false

```
{
    "Operator": "EQUALS",
    "Operands": ["vip"],
    "ComparisonValue": "$.Case.Fields.custom_case_field_id",
    "Negate": false
}
```

Case template condition

Parameters

- Operator - “CONTAINS_ANY”
- Operands – A list of template ids.
- ComparisonValue – "$.Case.TemplateId"
- Negate - true/false

```
{
"Operator": "CONTAINS_ANY",
"Operands": ["templateId"],
"ComparisonValue": "$.Case.TemplateId",
"Negate": false
}
```

OnSlaBreach

Cases SLA name condition

Parameters

- Operator - “CONTAINS_ANY”
- Operands – A list of SLA names.
- ComparisonValue – "$.RelatedItem.SlaConfiguration.Name"
- Negate - false

```
{
"Operator": "CONTAINS_ANY",
"Operands": ["highPrioritySla"],
"ComparisonValue": "$.RelatedItem.SlaConfiguration.Name",
"Negate": false
}
```

PatternMatch Operands

Each operand is an array of Pattern match objects.

PatternMatch Object

Type

- Description: The type of the pattern match object.
- Type: String
- Valid values: `PLAIN` | `LIST` | `PROXIMITY` | `NUMERICAL`
- Required: Yes

Value

- Description: Depending on the type, value type varies.
  - If type is `PLAIN`, Value is a string.
  - If type is `LIST`, Value is an array of `PLAIN` PatternMatch object.

- If type is PROXIMITY, Value is an object for format.

```
{
  "Distance": number,
  "IsWithin": boolean
}
```

- If type is NUMERICAL, Value is an object for format.

```
{
  "Decimal": boolean
}
```

Connect Customer testing language

This section describes the Connect Customer Testing language and how to use it. The Testing language is a JSON-based representation of testing observation, events and, actions.

Use the Testing language to do the following:

- Create test cases for simulating your flows.
- Write test cases instead of using the test designer.

Connect Customer Testing language concepts

The following terms are used in the Testing language.

Observations

Observations represent each complete interaction that includes one observed event expected from the system and many actions to validate or simulate system behaviors.

Events

Events represent expected behaviors that would come from the system, such as a prompt, a bot message, or a Lambda call.

Actions

Actions represent what the testing framework should do in response to an event, such as sending DTMF, responding with text, asserting attribute values, or ending the test.

Actors

Actors represent roles to be played in the testing framework. When observing events, actors can be the system or agent, such as a play prompt coming from the system or an agent accepting the call. When simulating actions, actors can be the customer, system, or agent, such as simulating a customer input DTMF or utterance, or simulating a system response from a Lambda function.

Example test

The following example demonstrates a JSON string representing the content of a test case configuration. It illustrates a simple test case that mocks the hours of operation when the test is initiated and then validates that a welcome prompt is played. Additionally, it simulates a contact's DTMF input to be placed in a queue and connected to an agent. The test will then verify if the queue placement is correct before concluding the test case. Overall, this test case configuration consists of three interactions with three simulated actions. Following is an example of the JSON representation for this test case.

```
{
    "Version": "2019-10-30",
    "Metadata": {},
    "Observations": [
        {
            "Identifier": "TriggerHoursCheck",
            "Event": {
                "Identifier": "Unique identifer"
                "Type": "TestInitiated",
                "Actor": "System",
                "Properties": {}
            },
            "Usage": {
                "Type": "EXACTLY"
            },
            "Actions": [
                {
                    "Identifier": "ActionId",
                    "Type": "OverrideSystemBehavior",
                    "Parameters": {
                        "ActionType": "OverrideSystemBehavior",
                        "Behavior": {
                            "Type": "FlowAction",
                            "Properties": {
                                "ActionType": "CheckHoursOfOperation",
                                "ActionParameters": {
                                    "HoursOfOperationId": "arn:aws:connect:us-
west-2:123456789012:instance/abc123/flow/BasicHours"
                                },
                                "Strategy": {
                                    "Type": "SubstituteResource",
```

```
                                    "SubstituteArn": "arn:aws:connect:us-
west-2:123456789012:instance/abc123/flow/AlwaysOnHours"
                                }
                            }
                        }
                    },
                    "Transitions": {
                        "NextAction": ""
                    }
                }
            ],
            "Transitions": {
                "NextObservations": [
                    "Welcome Message"
                ]
            }
        },
        {
            "Identifier": "Welcome Message",
            "Event": {
                "Identifier": "Unique identifer"
                "Type": "MessageReceived",
                "Actor": "System",
                "Properties": {
                    "Text": "Press 1 to be connected to an agent"
                },
                "MatchingCriteria": "Similarity"
            },
            "Usage": {
                "Type": "EXACTLY"
            },
            "Actions": [
                {
                    "Identifier": "Send dtmf input",
                    "Type": "SendInstruction",
                    "Actor": "Customer",
                    "Parameters": {
                        "ActionType": "SendInstruction",
                        "Actor" : "Customer",
                        "Instruction": {
                            "Type": "DtmfInput",
                            "Properties": {
                                "Value": 1
                            }
```

```
                        }
                    },
                    "Transitions": {
                        "NextAction": ""
                    }
                }
            ],
            "Transitions": {
                "NextObservations": [
                    "Trigger Queue Announcement"
                ]
            }
        },
        {
            "Identifier": "Trigger Queue Announcement",
            "Event": {
                "Type": "MessageReceived",
                "Actor": "System",
                "Properties": {
                    "Text": "Thank you for calling. Your call is very important to us
and will be answered in the order it was received."
                },
                "MatchingCriteria": "Similarity"
            },
            "Usage": {
                "Type": "EXACTLY"
            },
            "Actions": [
                {
                    "Identifier": "assertqueue",
                    "Type": "Assert",
                    "Parameters": {
                        "Namespace": "$.Queue.Name",
                        "Operator": "Equals",
                        "Operand": "Basic Queue"
                    },
                    "Transitions": {
                        "NextAction": "endthetest"
                    }
                },
                {
                    "Identifier": "endthetest",
                    "Type": "TestControl",
                    "Parameters": {
```

```
                        "ActionType": "TestControl"
                        "Command": {
                            "Type": "EndTest"
                        }
                    },
                    "Transitions": {
                        "NextAction": ""
                    }
                }
            ],
            "Transitions": {
                "NextObservations": []
            }
        }
    ]
}
```

Observations in the Connect Customer Testing language

Observations represent each complete interaction that includes one observed event expected from the system and many actions to validate or simulate system behaviors.

Parameters

- Version - The API version for the testing language, such as 2019-10-30.
- Metadata - Optional object containing UI-specific or non-functional data
- Observations - An array of observation objects that define the test flow

Observation object

Each observation consists of an event to observe and actions to execute when that event occurs.

- Identifier - Unique identifier for the observation
- Event - Defines the expected event from the system to observe
- Actions - Array of actions to execute when the event is observed
- Usage - Defines how many times this observation should be matched
  - Type: "EXACTLY"

  - Times: Integer value for the count, when applicable
- Transitions - Optional object defining flow control to next observations
  - NextObservations: Array of observation IDs to transition to

```
{
  "Version": "2019-10-30",
  "Metadata": { ... }, // Metadata to be used for data which is used for UI or any non-
runtime impacting data as required.
  "Observations": [
    {
      "Identifier": "Unique identifer",
      "Event": { ... },
      "Actions": [
            {
                "Identifier": "ActionId",
                "Type": "ActionType", // Action type could be of any type mentioned in
 recap (ObserveEvent, SendInstruction, Assertion, OverrideSystemBehavior, EndTest)
                "Parameters": {...},
                "Transitions" : {...}
            },
            ...
        ],
      "Usage": { "Type": "ANY" },
      "Transitions" : {
        "NextObservations": ["string-id", "string-id", "string-id"]
      }
    },
    // Additional observations...
  ]
}
```

Events in the Connect Customer Testing language

Events represent expected behaviors from the system that the test framework observes.

Contents

- Flow action started

Test initiated

Triggered when the test execution begins. This is typically used to set up initial conditions such as override system behaviors before the actual flow execution starts.

Parameters

- Identifier - Unique identifier for the event. (API need to specify this identifier in order for the UI to render properly)
- Type - Must always be `TestInitiated`.
- Actor - Must always be `System`. This indicates that the event originates from the testing system.
- Properties - Empty object. No additional properties are required.

```
{
    "Identifier": "unique identifier",
    "Type": "TestInitiated",
    "Actor": "System",
    "Properties": {}
}
```

Test completed

Triggered when the test execution ends. This event is observed when the test is terminated.

Parameters

- Type - Must always be `TestCompleted`.
- Actor - Must always be `System`. This indicates that the event originates from the testing system.
- Properties - Empty object. No additional properties are required.

```
{
    "Identifier": "unique identifier",
    "Type": "TestCompleted",
    "Actor": "System",
```

```
    "Properties": {}
}
```

Message received

Observes when the system plays a prompt or send any voice response to the simulated customer. This event can match messages using different criteria.

Parameters

- Identifier - Unique identifier for the event (API need to specify this identifier in order for the UI to render properly)
- Type - Must always be `MessageReceived`.
- Actor - Must always be `System`. This indicates the event originate from the testing system.
- Properties - Empty object. Object containing the following message details:
  - PromptId (Optional): Specific prompt ID or ARN to match
  - Text (Optional): Text content of the message to match
  - SSML (Optional): SSML content to match
  - Media (Optional): External media source details
    - Uri: Location of the media file
    - SourceType: Source from which media is fetched such as S3
    - MediaType: Type of media such as "audio/mpeg"
  - MatchingCriteria - Defines how to match the message:
    - Similarity: Uses semantic matching to find similar messages
    - Inclusion: Checks if the observed message contains the specified text

```
{
    "Identifier": "unique identifier",
    "Type": "MessageReceived",
    "Actor": "System",
    "Properties": {
        "PromptId": "string", // [Optional] A prompt ID or prompt ARN to play to
 the participant along with gathering input. May not be specified if Text or SSML is
 also specified. Must be specified either statically or as a single valid JSONPath
 identifier.
```

```
        "Text":  "string", // An optional string that defines text to send to the
 participant along with gathering input. May not be specified if PromptId or SSML is
 also specified. May be specified statically or dynamically.
        "SSML": "string", // An optional string that defines SSML to send to the
 participant along with gathering input. May not be specified if Text or PromptId is
 also specified May be specified statically or dynamically.
        "Media": { // An optional object that defines an external media source
            "Uri": "string", // Location of the message
            "SourceType": "string",// The source from which the message will be
 fetched. The only supported type is S3
            "MediaType": "string"// The type of the message to be played. The only
 supported type is Audio
        },
        "MatchingCriteria": { Type: Similarity / Inclusion }
    }
}
```

Flow action started

Observes when a specific flow action begins execution. This allows you to detect when particular flow actions are executed during the simulation.

Invoke Lambda function

Observes when a Lambda function invocation action starts.

Parameters

- Identifier - Unique identifier for the event (In preview, API need to specify this identifier in order for the UI to render properly)
- Type - Must always be `FlowActionStarted`.
- Actor - Must always be `System`. This indicates that the event originates from the testing system.
- Properties:
  - ActionType - Must always be `InvokeLambdaFunction`.
  - ActionParameters:
    - LambdaFunctionARN: The ARN of the Lambda function being invoked

```
{
    "Identifier": "unique identifier",
```

```
    "Type": "FlowActionStarted",
    "Actor": "System",
    "Properties": {
        "ActionType" : "InvokeLambdaFunction " // V2 JSON action type
        "ActionParameters" : { // V2 JSON action parameters
            "LambdaFunctionARN": "string"
        }
}
```

Check hours of operation

Observes when the flow checks hours of operation.

Parameters

- Identifier - Unique identifier for the event (API need to specify this identifier in order for the UI to render properly)
- Type - Must always be `FlowActionStarted`.
- Actor - Must always be `System`. This indicates that the event originates from the testing system.
- Properties:
  - ActionType - Must always be `CheckHoursOfOperation`.
  - ActionParameters:
    - HoursOfOperationId: The ID or ARN of the hours of operation resource being checked

```
{
    "Identifier": "unique identifier",
    "Type": "FlowActionStarted",
    "Actor": "System",
    "Properties": {
        "ActionType" : "CheckHoursOfOperation" // V2 JSON action type
        "ActionParameters" : { // V2 JSON action parameters
            "HoursOfOperationId": "string"
        }
}
```

Transfer contact to queue

Observes when a contact is being transferred to a queue.

Parameters

- Identifier - Unique identifier for the event (API need to specify this identifier in order for the UI to render properly)
- Type - Must always be `FlowActionStarted`.
- Actor - Must always be `System`. This indicates that the event originates from the testing system.
- Properties:
  - ActionType - Must always be `TransferContactToQueue`.
  - ActionParameters:
    - QueueId: The ID or ARN of the target queue
    - AgentId (Optional): Specific agent ID if transferring to a particular agent

```
{
    "Identifier": "unique identifier",
    "Type": "FlowActionStarted",
    "Actor": "System",
    "Properties": {
        "ActionType" : "TransferContactToQueue" // V2 JSON action type
        "ActionParameters" : { // V2 JSON action parameters
            "QueueId": "string",
            "AgentId" : "string"
        }
}
```

Connect participant with Lex bot

Observes when a participant is being connected to a Lex bot.

Parameters

- Identifier - Unique identifier for the event (API need to specify this identifier in order for the UI to render properly)
- Type - Must always be `FlowActionStarted`.
- Actor - Must always be `System`. This indicates that the event originates from the testing system.
- Properties:
  - ActionType - Must always be `ConnectParticipantWithLexBot`.

- ActionParameters:
  - LexV2Bot: Object containing bot details
    - AliasArn: The ARN of the Lex V2 bot alias

```
{
    "Identifier": "unique identifier",
    "Type": "FlowActionStarted",
    "Actor": "System",
    "Properties": {
        "ActionType": "ConnectParticipantWithLexBot"
        "ActionParameters": {
            "LexV2Bot": {
                "AliasArn": "string"
            }
        }
    }
}
```

Actions in the Connect Customer Testing language

Actions represent what the testing framework executes in response to observed events.

Assertion

Validates that a specific attribute or value in the flow matches expected criteria.

Parameters

- Identifier: Unique identifier for the action
- Type: Must be `Assert`
- Parameters:
  - Namespace: JSON path to the attribute value to check
  - Operator: Comparison operator to use (see Supported operators)
  - Operand: Expected value to compare against
- Transitions
  - NextAction: The unique identifier for the next action

Supported operators

- Equals: Exact match
- TextStartsWith: Text begins with the operand
- TextEndsWith: Text ends with the operand
- TextContains: Text contains the operand
- NumberGreaterThan: Numeric value is greater than operand
- NumberGreaterOrEqualTo: Numeric value is greater than or equal to operand
- NumberLessThan: Numeric value is less than operand
- NumberLessOrEqualTo: Numeric value is less than or equal to operand
- Exists: Checks if the attribute exists (operand not required)

```
{
   "Identifier" : "ActionId",
   "Type" : "Assert",
   "Parameters" : {
       "Namespace" : "string", // Data path to fetch
       "Operator": "string", // Comparasion operator
       "Operand" : "string" // Expected Value to compare
   }
    "Transitions": { "NextAction": "string" }
}
```

Send Instruction

Simulates customer input during the test execution. This action allows you to send DTMF tones or voice input as if a customer were interacting with the flow.

DTMF input

Simulates a customer pressing keys on their phone keypad.

Parameters

- Identifier: Unique identifier for the action
- Type: Must be `SendInstruction`
- Parameters:

- ActionType: Must be `SendInstruction`
- Actor: `Customer` indicates this simulates customer behavior
- Instruction: Object defining the instruction type
  - Type: Must be `DtmfInput`
  - Properties:
    - Value: String or number representing the DTMF input (e.g., "1", "#", "*")
- Transitions:
  - NextAction: The unique identifier for the next action

```
{
    "Identifier": "ActionId",
    "Type": "SendInstruction",
    "Parameters": {
        "ActionType": "SendInstruction",
        "Actor" : "Customer",
        "Instruction": {
            "Type": "DtmfInput",
            "Properties": {
                "Value": "string"
            }
        }
    },
    "Transitions": { "NextAction": "string" }
}
```

Text and Utterance Input

Simulates customer voice or text input. This is used for Lex bot or AI agent interactions.

Parameters

- Identifier: Unique identifier for the action
- Type: Must be `SendInstruction`
- Parameters:
  - ActionType: Must be `SendInstruction`
  - Actor: `Customer` indicates this simulates customer behavior
  - Instruction: Object defining the instruction

  - Type: Must be `TextUtterance`
  - Properties:
    - Text (Optional): Plain text input to send
    - SSML (Optional): SSML-formatted input to send
    - LanguageCode: Language code for the input (e.g., "en-US")
- Transitions:
  - NextAction: The unique identifier for the next action

```
{
    "Identifier": "ActionId",
    "Type": "SendInstruction",
    "Parameters": {
        "ActionType": "SendInstruction",
        "Actor" : "Customer",
        "Instruction": {
            "Type": "TextUtterance",
            "Properties": {
                "Text":  "string", // An optional string that defines text to send to
 the participant along with gathering input. May not be specified if PromptId or SSML
 is also specified. May be specified statically or dynamically.
                "SSML": "string", // An optional string that defines SSML to send to
 the participant along with gathering input. May not be specified if Text or PromptId
 is also specified May be specified statically or dynamically.,
                "LanguageCode": "en-US"
            }
        }
    },
    "Transitions": { "NextAction": "string" }
}
```

Disconnect

Simulates customer ending the call.

Parameters

- Identifier: Unique identifier for the action
- Type: Must be `SendInstruction`

- Parameters:
  - ActionType: Must be `SendInstruction`
  - Actor: `Customer` indicates this simulates customer behavior
  - Instruction: Object defining the instruction
    - ActionType: Must be `SendInstruction`
    - Type: Must be `Disconnect`
- Transitions:
  - NextAction: The unique identifier for the next action

```
{
    "Identifier": "ActionId",
    "Type": "SendInstruction",
    "Parameters": {
        "ActionType": "SendInstruction",
        "Actor" : "Customer",
        "Instruction": {
            "Type": "Disconnect",
        }
    },
    "Transitions": { "NextAction": "string" }
}
```

Override system behavior

Modifies how specific flow actions behave during test execution. This allows you to mock external dependencies, substitute resources, or simulate specific scenarios without modifying the actual flow.

InvokeLambdaFunction

Override Lambda function behavior by substituting with a different function or mocking the response.

Substitute resource strategy

Redirects Lambda invocations to a different function ARN.

Parameters

- Identifier: Unique identifier for the action
- Type: Must be `OverrideSystemBehavior`
- Parameters:
  - ActionType: Must be `OverrideSystemBehavior`
  - Behavior: Object defining the behavior to override
    - Type: Must be `FlowAction`
    - Properties:
      - ActionType: Must be `InvokeLambdaFunction`
      - ActionParameters:
        - LambdaFunctionARN: The ARN of the Lambda function to override
      - Strategy: Object defining the override strategy
        - Type: Must be `SubstituteResource`
        - SubstituteArn: ARN of the replacement Lambda function to use
- Transitions:
  - NextAction: The unique identifier for the next action

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType" : "InvokeLambdaFunction",
                "ActionParameters": {
                    "LambdaFunctionARN" : "string"
                },
                "Strategy": {
                    "Type": "SubstituteResource",
                    "SubstituteArn": "string"
                }
            }
        }
```

```
    },
    "Transitions": { "NextAction": "string" }
}
```

Mock response strategy - Success

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType" : "InvokeLambdaFunction",
                "ActionParameters": {
                    "LambdaFunctionARN" : "string"

                },
                "Strategy": {
                    "Type": "MockResponse",
                    "Response": {
                        "Type" : "ExecutionResult",
                        "ExecutionResult" : {
                            "DelaySeconds" : Number,
                            "LoadedData" : "serialized JSON"
                        }
                    }
                }
            }
        }
    },
    "Transitions": { "NextAction": "string" }
}
```

Mock response strategy - Error

Simulates a Lambda function error without actual invocation.

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
```

```
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType" : "InvokeLambdaFunction",
                "ActionParameters": {
                    "LambdaFunctionARN" : "lambda-arn-to-mock"
                },
                "Strategy": {
                    "Type": "MockResponse",
                    "Response": {
                        "Type" : "Error",
                        "Error" : {
                            "DelaySeconds" :  Number,
                            "Value" : "TimeLimitExceeded|NoMatchingError"
                        }
                    }
                }
            }
        },
        "Transitions": { "NextAction": "string" }
}
```

CheckHoursOfOperation

Override hours of operation checks to test different time-based scenarios.

Substitute resource strategy

Redirects hours of operation checks to a different hours of operation configuration.

Parameters

- Identifier - Unique identifier for the action
- Type - Must be `OverrideSystemBehavior`
- Parameters:
  - ActionType: Must be `OverrideSystemBehavior`
  - Behavior: Object defining the behavior to override
    - Type: Must be `FlowAction`
    - Properties

        - ActionType: Must be `CheckHoursOfOperation`
        - ActionParameters:
            - HoursOfOperationId: The ID/ARN of the hours of operation to override
        - Strategy:
            - Type: Must be `SubstituteResource`
            - SubstituteArn: ARN of the replacement hours of operation resource
- Transitions:
    - NextAction: The unique identifier for the next action

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType" : "CheckHoursOfOperation",
                "ActionParameters": {
                    "HoursOfOperationId" : "string"
                },
                "Strategy": {
                    "Type": "SubstituteResource",
                    "SubstituteArn": "string"
                }
            }
        }
    },
    "Transitions": { "NextAction": "string" }
}
```

Mock response strategy - Success

Returns a predefined hours of operation check result.

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
```

```
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType": "CheckHoursOfOperation",
                "ActionParameters": {
                    "HoursOfOperationId": "string"
                },
                "Strategy": {
                    "Type": "MockResponse",
                    "Response": {
                        "Type": "ExecutionResult",
                        "ExecutionResult": {
                            "Value": "InHours|OutOfHours"
                        }
                    }
                }
            }
        },
        "Transitions": {
            "NextAction": "string"
        }
    }
}
```

Mock response strategy - Error

Simulates an error during hours of operation check.

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType": "CheckHoursOfOperation",
                "ActionParameters": {
                    "HoursOfOperationId": "arn:aws:connect:us-
west-2:489887583805:instance/0b6b36d4-add8-4fc2-bb5b-72b8e4042838/operating-
hours/90d61ae9-79b5-4940-91e5-3e38f8383958"
                },
                "Strategy": {
```

```
                    "Type": "MockResponse",
                    "Response": {
                        "Type": "Error",
                        "Error": {
                            "Value": "NoMatchingError"
                        }
                    }
                }
            }
        }
    },
    "Transitions": {
        "NextAction": "string"
    }
}
```

ConnectParticipantWithLexBot

Override Lex bot behaviors to use a different bot for testing or mock responses.

Substitute resource strategy

Parameters

- Identifier - Unique identifier for the action
- Type - Must be `OverrideSystemBehavior`
- Parameters
  - ActionType - Must be `OverrideSystemBehavior`
  - Behavior:
    - Type: Must be `FlowAction`
    - Properties:
      - ActionType - Must be `ConnectParticipantWithLexBot`
      - ActionParameters:
        - LexV2Bot: Object containing the bot to override
          - AliasArn: ARN of the Lex bot alias to override
      - Strategy:
        - Type: Must be `SubstituteResource`
        - SubstituteArn: ARN of the replacement Lex bot alias

- Transitions:
  - NextAction: The unique identifier for the next action

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType" : "ConnectParticipantWithLexBot",
                "ActionParameters" : {
                    "LexV2Bot": {
                       "AliasArn": "string"
                    }
                },
                "Strategy": {
                    "Type": "SubstituteResource",
                    "SubstituteArn": "string"
                }
             }
         }
    },
    "Transitions": { "NextAction": "string" }
}
```

Mock response strategy - Success

Returns a predefined successful response without invoking the actual Lex bot.

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType": "ConnectParticipantWithLexBot",
                "ActionParameters": {
                    "LexV2Bot": {
```

```
                        "AliasArn": "string"
                    }
                },
                "Strategy": {
                    "Type": "MockResponse",
                    "Response": {
                        "Type": "ExecutionResult",
                        "ExecutionResult": {
                            "DelaySeconds": Number,
                            "LoadedData": "serialized JSON"
                        }
                    }
                }
            }
        }
    },
    "Transitions": { "NextAction": "string" }
}
```

Mock response strategy - Error

Simulates a Lex bot with error without actual invocation.

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType": "ConnectParticipantWithLexBot",
                "ActionParameters": {
                    "LexV2Bot": {"AliasArn": "string"}
                },
                "Strategy": {
                    "Type": "MockResponse",
                    "Response": {
                        "Type": "Error",
                        "Error": {
                            "DelaySeconds": Number,
                            "Value": "TimeLimitExceeded|NoMatchingError"
                        }
                    }
```

```
                }
            }
        }
    },
    "Transitions": { "NextAction": "string" }
}
```

TransferContactToQueue

Override queue transfer behavior to substitute queues or simulate transfer failures.

Substitute resource strategy

Redirects queue transfers to a different queue.

Parameters

- Identifier - Unique identifier for the action
- Type - Must be `OverrideSystemBehavior`
- Parameters
  - ActionType - Must be `OverrideSystemBehavior`
  - Behavior
    - Type: `FlowAction`
    - Properties
      - ActionType - Must be `TransferContactToQueue`
      - ActionParameters:
        - QueueId: ID/ARN of the queue to override
      - Strategy:
        - Type: Must be `SubstituteResource`
        - SubstituteArn: ARN of the replacement queue
- Transitions:
  - NextAction: The unique identifier for the next action

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
```

```
    "Parameters": {
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType" : "TransferContactToQueue",
                "ActionParameters" : {
                    "QueueId": "string"
                },
                "Strategy": {
                    "Type": "SubstituteResource",
                    "SubstituteArn": "string"
                }
             }
         }
    },
    "Transitions": { "NextAction": "string" }
}
```

Mock response strategy - Error

Simulates queue transfer failures for testing error paths.

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType": "TransferContactToQueue",
                "ActionParameters": {
                    "QueueId": "string"
                },
                "Strategy": {
                    "Type": "MockResponse",
                    "Response": {
                        "Type": "Error",
                        "Error": {
                            "Value": "QueueAtCapacity|NoMatchingError"
                        }
                    }
                }
```

```
            }
        }
    },
    "Transitions": {
        "NextAction": "string"
    }
}
```

DequeueAndTransferToQueue

Override behavior when dequeuing a contact and transferring to another queue.

Substitute resource strategy

Parameters

- Identifier - Unique identifier for the action
- Type - Must be `OverrideSystemBehavior`
- Parameters
  - ActionType - Must be `OverrideSystemBehavior`
  - Behavior:
    - Type: `FlowAction`
    - Properties:
      - ActionType - Must be `DequeueAndTransferToQueue`
      - ActionParameters:
        - QueueId: ID/ARN of the queue to override
      - Strategy:
        - Type: Must be `SubstituteResource`
        - SubstituteArn: ARN of the replacement queue
- Transitions
  - NextAction: The unique identifier for the next action

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
```

```
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType" : "DequeueAndTransferToQueue",
                "ActionParameters" : {
                    "QueueId": "string"
                },
                "Strategy": {
                    "Type": "SubstituteResource",
                    "SubstituteArn": "string"
                }
             }
         }
    },
    "Transitions": { "NextAction": "string" }
}
```

Mock response strategy - Error

Simulates dequeue and transfer failures.

```
{
    "Identifier": "ActionId",
    "Type": "OverrideSystemBehavior",
    "Parameters": {
        "ActionType": "OverrideSystemBehavior",
        "Behavior": {
            "Type": "FlowAction",
            "Properties": {
                "ActionType": "DequeueAndTransferToQueue",
                "ActionParameters": {
                    "QueueId": "string"
                },
                "Strategy": {
                    "Type": "MockResponse",
                    "Response": {
                        "Type": "Error",
                        "Error": {
                            "Value": "QueueAtCapacity|NoMatchingError"
                        }
                    }
                }
            }
```

```
        }
    },
    "Transitions": {
        "NextAction": "string"
    }
}
```

Test Control Actions

Special actions that control test execution flow and debugging.

End Test

Explicitly terminates the test execution. Use this when you want to end the test at a specific point rather than waiting for natural completion.

Parameters

- Identifier - Unique identifier for the action
- Type - Must be `TestControl`
- Parameters:
  - Action Type: Must be `TestControl`
  - Command: Object defining the control command
    - Type: Must be `EndTest`
- Transitions
  - NextAction: The unique identifier for the next action

```
{
    "Identifier": "ActionId",
    "Type": "TestControl",
    "Parameters": {
        "ActionType": "TestControl"
        "Command": {
           "Type": "EndTest"
        }
    },
    "Transitions": { "NextAction": "string" }
}
```

Log Data

Captures and logs specific attribute values during test execution for debugging and validation purposes. The logged data appears in the test execution results.

Parameters

- Identifier - Unique identifier for the action
- Type - Must be `TestControl`
- Parameters:
  - Action Type: Must be `TestControl`
  - Command: Object defining the control command
    - Type: Must be `LogData`
    - Properties:
      - Expressions: Object containing key-value pairs where:
        - Key: A descriptive label for the logged value
        - Value: JSON path to extract the attributes (e.g., "$.Queue.Name")
- Transitions
  - NextAction: The unique identifier for the next action

```
{
    "Identifier": "ActionId",
    "Type": "TestControl",
    "Parameters": {
        "ActionType": "TestControl",
        "Command": {
            "Type": "LogData",
            "Properties": {
                "Expressions": {
                    "string":"string", //"myContactId": "$.ContactId"
                ...
                }
            }
        }
    },
    "Transitions": { "NextAction": "string" }
}
```

Document history for the Amazon Connect Developer Guide

The following table describes the documentation releases for Amazon Connect.

Change	Description	Date
Initial release	Initial release of the Amazon Connect Developer Guide. This guide provides developer-focused content including API best practices, the Connect flow language, rules function language, and testing language.	May 15, 2026