

Installation and Initial Configuration guide

Overview

This component helps you quickly establish connection between Pega and NICE inContact, as cloud-based telephony software to provide the CSR's a seamless experience of login in to the MAX Dialer, maintaining agent state between NICE inContact and Pega Customer Service and accept incoming calls with call coordinates for customer verification and customer 360.

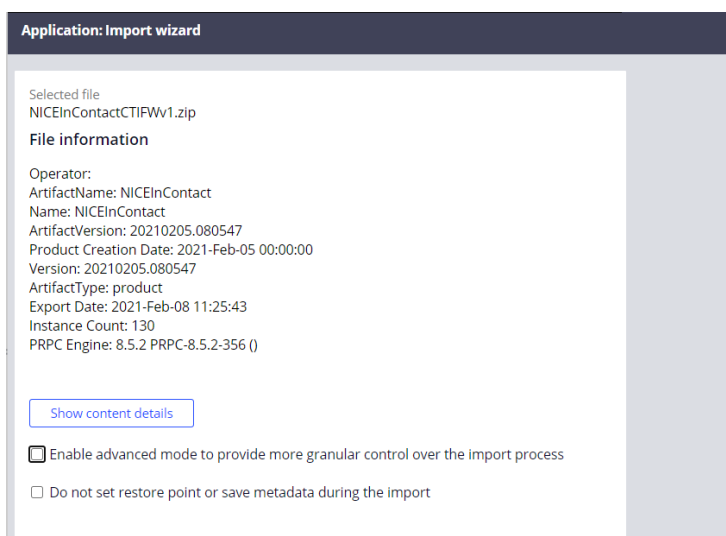
Being a server-side Open CTI implementation, the elementary architecture of the communication between Pega Call and NICE inContact is based on REST API calls. These API's are cloud based and can be invoked from Pega with help of an authorization header consisting of an accessKeyId and accessKeySecret.

All the other API's starting from getting the Agent ID based on the email address to joining an existing session using that Agent ID to getting the agent state to getting the next event; all of them will use this authorization header for access.

NICE inContact uses a typical "continuous poll-based" architecture to get the next set of events as far as agent state or call contact event is concerned. For this purpose, they have an API named "GetNextEvent" which gives you a list of events that happened for the last 'n' seconds where 'n' is a DSS that we have defined in Pega. As part of this implementation, we have designed two queue processors which help in orchestrating these events to the Pega Call event handlers and that in turn manages the agent states and the call pop-up in the Interaction Portal.

The below steps describe the basic steps to perform post deployment for a successful test.

1. Upload the provided zip file using Pega Import wizard



The screenshot shows the 'Application: Import wizard' window. It displays the selected file 'NICEInContactCTIFW1.zip' and its file information. The information includes the operator, artifact name, version, product creation date, artifact type, export date, instance count, and PRPC engine version. There is a 'Show content details' button and two checkboxes for advanced mode and restore point settings.

Application: Import wizard

Selected file
NICEInContactCTIFW1.zip

File information

Operator:
ArtifactName: NICEInContact
Name: NICEInContact
ArtifactVersion: 20210205.080547
Product Creation Date: 2021-Feb-05 00:00:00
Version: 20210205.080547
ArtifactType: product
Export Date: 2021-Feb-08 11:25:43
Instance Count: 130
PRPC Engine: 8.5.2 PRPC-8.5.2-356 ()

Show content details

☒ Enable advanced mode to provide more granular control over the import process



☐ Do not set restore point or save metadata during the import

2. The product file contains

- a. an application rule “NICEFW-CTI:01.01.01” which can either be used as a “Built on application” in your existing application or

Built on applications



[+ Add application](#)

	Name	Version	
1	NICEFW-CTI	01.01.01	 

- b. a ruleset “NICE-CTI:01-01-01” which can be directly used as an “Application ruleset” in your existing application

Application rulesets

[+ Add ruleset](#)

1	NICE-CTI:01-01	 
---	----------------	---

3. Post deployment check and update the following items

- a. Dynamic system settings
 - i. nice/AccessKeyId – This access key id will be used to fetch the access key token using D_GetAccessToken. Reach out to your NICE inContact representative to get the correct access key id.

Edit Dynamic System Settings: NICE Access Key ID

ID: NICE-CTI • nice/AccessKeyId RS: NICE-CTI [Edit]

[Settings](#) [History](#)

Value

Set your access key id here

- ii. nice/AccessKeySecret – This access key secret will be used to fetch the access key token using D_GetAccessToken. Reach out to your NICE inContact representative to get the correct access key secret.

Edit Dynamic System Settings: NICE Access Key Secret
ID: NICE-CTI • nice/AccessKeySecret RS: NICE-CTI [Edit]

Settings History

Value
Set your access key secret here

- iii. nice/AuthVersion – This auth version will be used in all of your Rest API connections to NICE inContact. Reach out to your NICE inContact representative to get the correct version. This version will be used for the following API calls.

1. AccessKey
2. GetUserList

Edit Dynamic System Settings: NICE Auth Version
ID: NICE-CTI • nice/AuthVersion RS: NICE-CTI [Edit]

Settings History

Value
Set your auth version here

- iv. nice/BaseVersion – This base version will be used in all of your Rest API connections to NICE inContact. Reach out to your NICE inContact representative to get the correct version. This version will be used for the following API calls.

1. GetAgentId
2. JoinAgentSession
3. GetAgentState
4. SetAgentState
5. LogoutAgent
6. GetAgentUnavailableCodes
7. GetNextEvent

Edit Dynamic System Settings: NICE Base Version
ID: NICE-CTI • nice/BaseVersion RS: NICE-CTI [Edit]

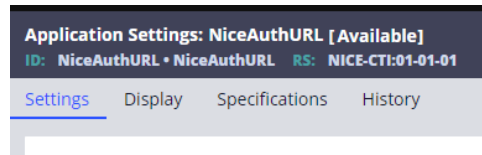
Settings History

Value
Set your base version here

b. Application Settings

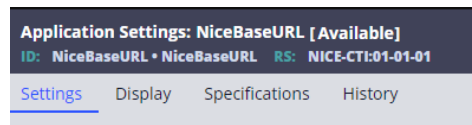
- i. NiceAuthURL – This is the Base URL (application setting being used for connections to following NICE inContact API's. Reach out to your NICE inContact representative to get the correct URL.

1. AccessKey
2. GetUserList

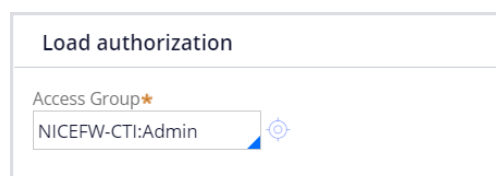


- ii. NiceBaseURL – This is the Base URL (application setting being used for connections to following NICE inContact API's. Reach out to your NICE inContact representative to get the correct URL.

1. GetAgentId
2. JoinAgentSession
3. GetAgentState
4. SetAgentState
5. LogoutAgent
6. GetAgentUnavailableCodes
7. GetNextEvent



- c. Change the access group for the following Node level data pages. Consider changing it to any application-level access group that you may have created for your application specific Node level data pages having access to the ruleset “NICE-CTI”. Currently this is set to a mock access group “NICEFW-CTI:Admin” pointing to application “NICEFW-CTI:01.01.01”. This step is applicable only if you are following step #2(b).



- i. D_NICEGlobalSettings
- ii. D_GetAccessToken
- iii. D_GetNiceUserList
- iv. D_AgentUnavailableCodes

4. If you follow the approach in #2(a) then ensure that the application “NICEFW-CTI” is added to your System Runtime Context. In case you follow the approach in #2(b) then ensure that the application where you have added the ruleset “NICE-CTI” as an application ruleset is added to your System Runtime Context. This will allow the two queue processors (a) GetNextEvent and (b) HandleEvent to run smoothly.
5. You may customize or extend the rules by creating a new version of “NICE-CTI” ruleset.
6. The implementation is based on the fact that the email address specified in the Operator ID instance of the customer representative will be used as the credential to join an existing session in NICE inContact. Hence the below pre-requisites are mandatory before you proceed with the testing
 - a. You need to have a profile in NICE inContact
 - b. The email address associated with the user profile in NICE inContact must be same as the email address mentioned in the Operator ID of the customer service representative.

- c. Before connecting to NICE inContact from Pega, you must have valid session in MAX Dialer by launching it from NICE inContact.

