



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

Composer Help

Integrated Voice Route Project Templates

12/15/2025

Integrated Voice Route Project Templates

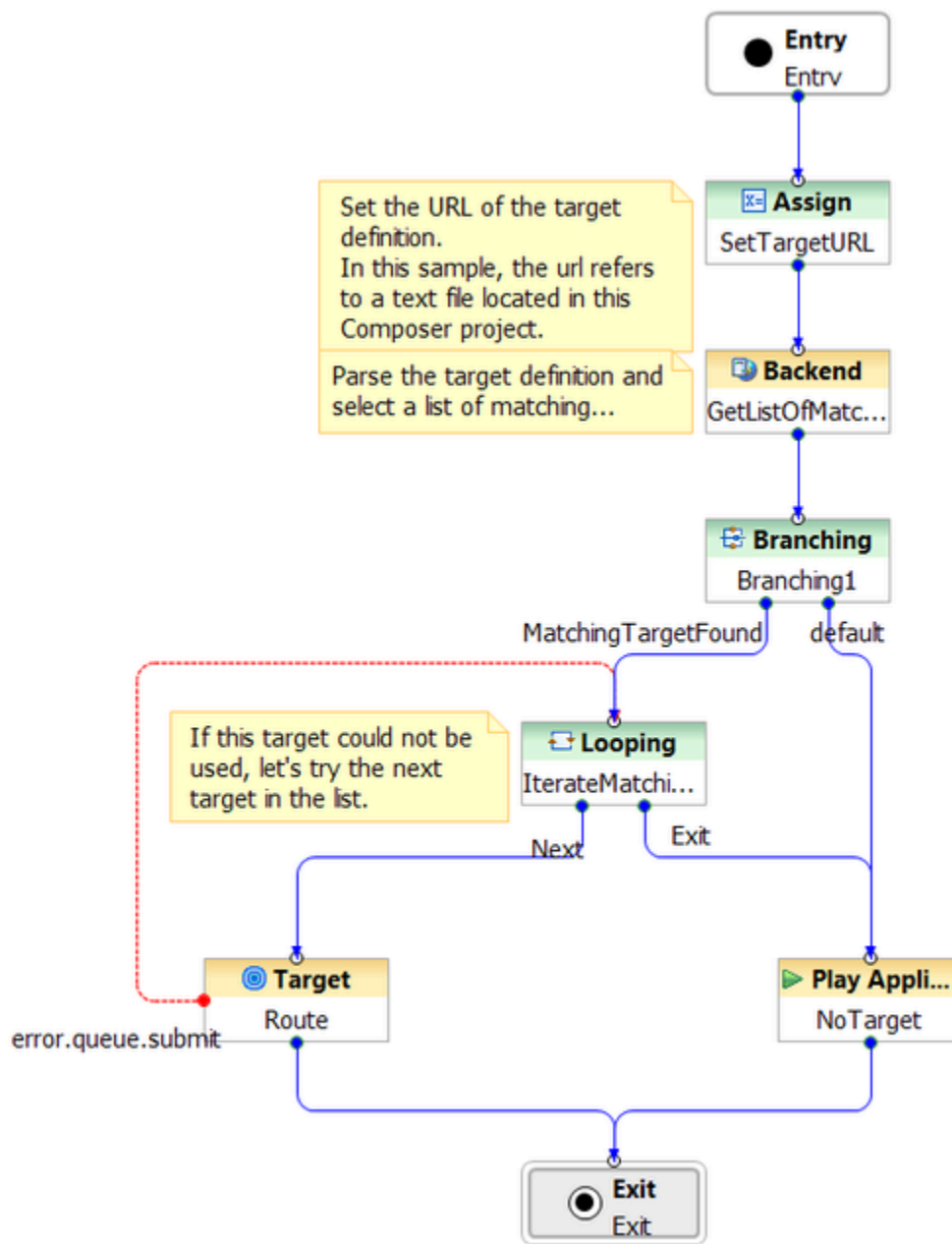
Contents

- **1 Integrated Voice Route Project Templates**
 - **1.1 External File-Based Routing Template**
 - **1.2 Load Balancing and Working Hour Routing Template**
 - **1.3 Play Application & Busy Treatment Template**
 - **1.4 Routing Based on DNIS and ANI Template**

You can create an Integrated Voice (**GVP**) and Routing (**Orchestration**) application based on a Project template. A few of the supplied samples are described below. Also see the video [Integrated Voice and Route Application](#).

External File-Based Routing Template

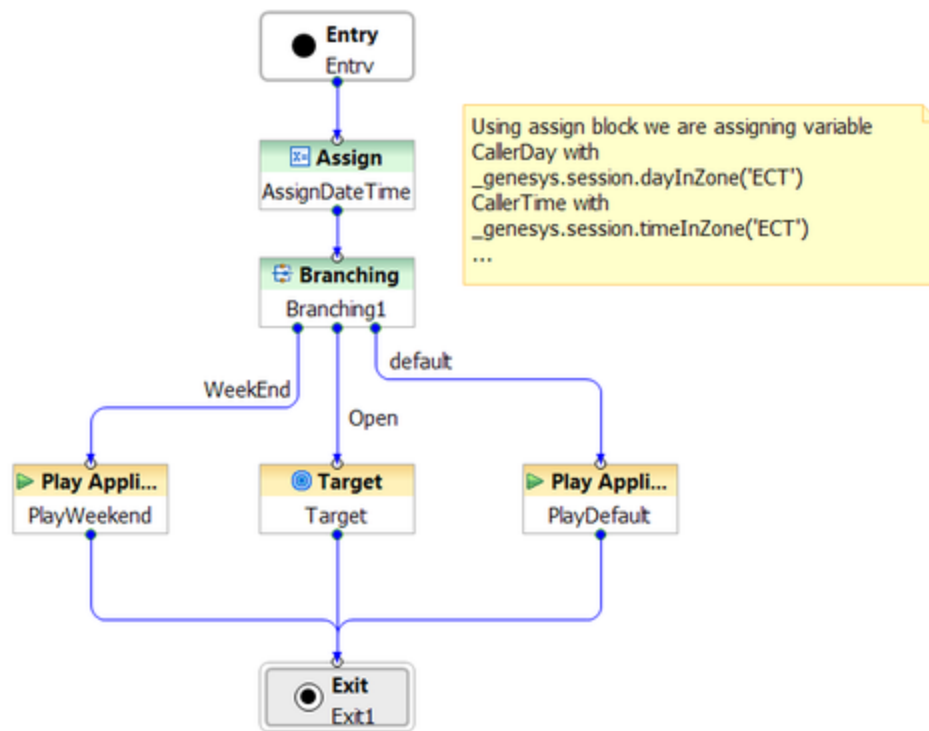
This template demonstrates voice call routing based on criteria contained in an external file.



1. The **IPD** contains a single Workflow block pointing to a workflow diagram named default.workflow.
2. The default.workflow diagram starts with an **Assign block** that sets the URL of the target definition. In this case, the target definition (routing criteria and target preferences) is defined in the target.txt file contained in the src folder of the Project (but which could be defined in a file external to the Project). Routing target preferences are based on criteria such as the number dialed (DNIS), whether the inquiry is about home or auto insurance, the caller's language, claim history, discount percentage, and so on. Calls will be routed to various agent groups/queues based this information, which is assumed to be contained in the user data of the voice interaction.
3. The next block in the default.workflow diagram is a **Backend block**, which is used for invoking custom backend Java Server Pages; in this case, to execute the routing logic. Using the target.txt file DNIS criteria, the Backend block parses the target definition and selects a list of matching targets that could be routed to.
4. A **Branching block** then segments interactions to take different paths in the workflow.
 - If a matching target is found, the Branching block sends the voice interaction to a Looping block, which iterates until an exit condition is met; in this case, when an available target is found, which could be an agent, agent group, or the result of a Skill Expression.
 - If repeated Looping block iterations do not find an available target, the voice interaction goes to a **PlayApplicationblock**, which could inform the caller that all agents are busy and to call back later.

Load Balancing and Working Hour Routing Template

The template is shown below. For a video tutorial on this template, see [Integrated Voice and Route Application](#).



The **Assign Block** assigns the following variables:

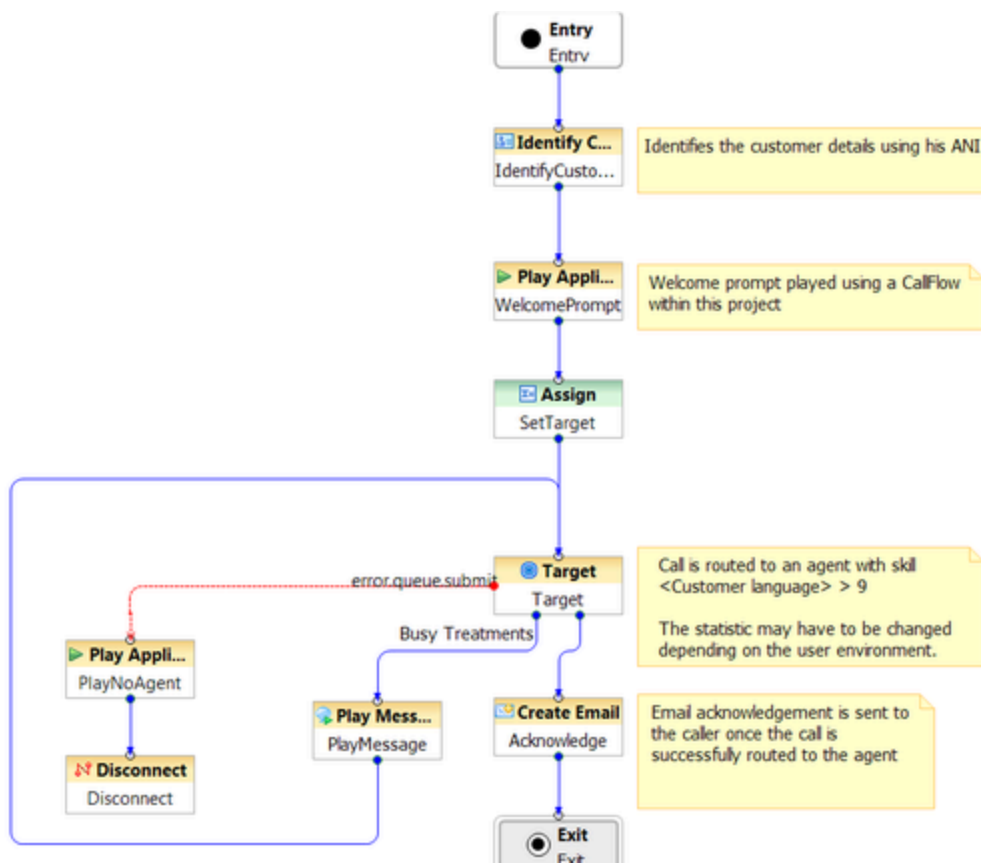
- CallerDay using the Orchestration Server function available in **ExpressionBuilder** called `_genesys.session.dayInZone('ECT')`
- CallerTime using the Universal Routing Server functions available in Expression Builder called `_genesys.session.timeInZone('ECT')`

Next, a **Branching block** checks for the following conditions:

- If CallerDay is Saturday or Sunday, then a **Play Application block** play the Weekend message to the caller.
- If CallerDay is a week day and CallerTime is after business hours, then a Play Application block plays the After Business Hours message to the caller.
- If CallerDay is a week day and CallerTime is within business hours, then a **Target block** is used to load balance calls between two agent groups.

Play Application & Busy Treatment Template

The template is shown below.

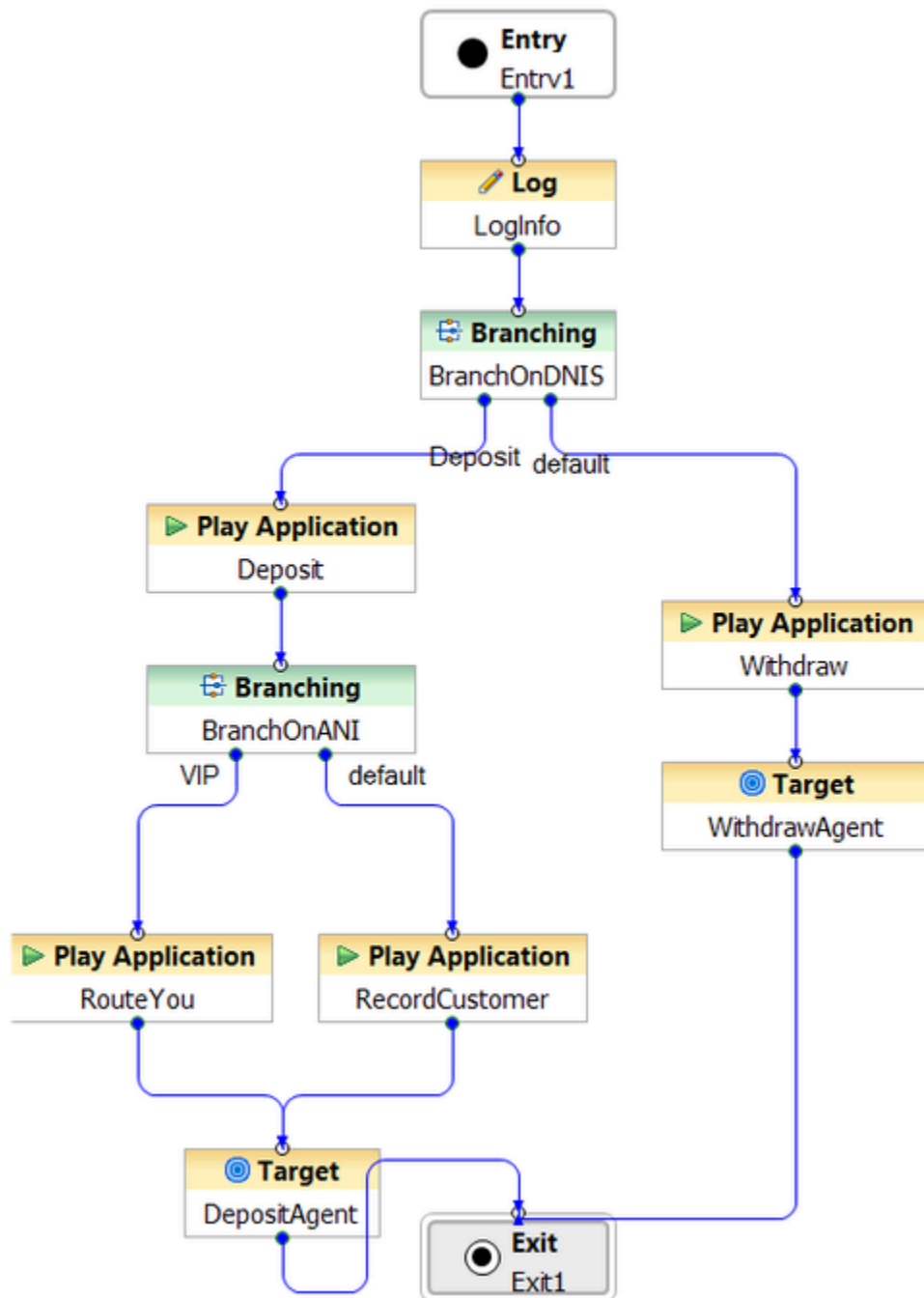


After the Entry block, this sample starts with an **Identify Customer** block that attempts to identify the calling customer in the Universal Contact Server Database based on the calling party's number stored in the ANI variable defined in the Entry block. The Customer Attributes property specifies to search/retrieve **core customer profile** attributes for the phone number.

1. A **Log** block logs customer information retrieved from the database. The Logging Details property specifies a JSON string for the format of the customer data.
2. A **Play Application** block plays a welcome message to the caller. The Language property indicates the customer's language is English. The Parameters property specifies the customer's first and last name from the JSON string.
3. A **Target** block routes the call based on a skill expression to a target with an English skill greater than 9 and a Spanish skill greater than 5.
4. If there are no available routing targets, a **Play Message** block plays a message to the caller. The Prompts property defines an expression (created in **Expression Builder**) that calls an **Orchestration Server function** (`_genesys.statistic.sData`). This function returns the value of a statistic (ExpectedWaitTime) for a specified object, in this case, a Skill Expression. This could be announced to the caller via an IVR application while waiting for a target.
5. Once the call is successfully routed to an agent, a **Create E-mail** block creates an acknowledgement e-mail to the customer in the form of pre-written text from the Standard Response library (see **eServices Help for Knowledge Manager**). The customer's e-mail address is contained in the customeremail variable defined in Entry block.

Routing Based on DNIS and ANI Template

The template is shown below.



This template demonstrates how the originating phone number (ANI) and dialed phone number (DNIS) values are automatically made available as variables when you create a workflow.

1. The BranchOnDNIS block first checks if the incoming call was made to the DNIS for making a deposit.
2. If the call was not made to the deposit DNIS, the application will play a voice treatment before routing to the withdraw agent. The Withdraw Play Application block points to a voice callflow in the Composer Project. This will execute a VXML page which plays a message to the caller. The WithdrawAgent Target block routes the call to the agent responsible for handling withdrawals.
3. If the DNIS did match the one for deposit, the application first plays an application treatment with a message for the caller. It will then examine the ANI of the caller in the BranchOnANI block, which contains the following branching condition: ANI==Customer. In this case, "Customer" is a variable declared in the Entry block of the workflow, and it can be modified to specify the ANI of the "special" customer that you are targeting.
4. The special customer is then routed directly to the agent handling deposits. Non-special customers must go through a voice treatment which records their name before proceeding. The voice treatment is executed by the RecordCustomer Play Application block.
5. A voice callflow is then executed by the RecordCustomer block. After recording the user's name, the Menu block provides the option to listen to the recording, re-record the audio, or exit. Upon exiting the voice treatment, the call will be routed to the deposit agent.