Genesys Events and Models Reference

Unified Call-Party States

12/16/2025

Unified Call-Party States

Contents

A *call-party* is the relationship between a call and a given telephony device. Call-Party is often referred to as party. For the purposes of this section, the two terms are interchangeable.

A *call* is typically an interaction between two or more telephony objects (or between a telephony object and a network entity) that is established by the use of telephony network capabilities. However, in fact, Genesys assumes this association may have zero to many parties. In the context of the Genesys model, a call is a stateless object, and it always has a unique connection ID.

A *party* (call-party) represents the call's relationship to a given telephony device—either an internal DN or an external (out-of-PBX) call participant. Each call-party has state (and a number of other attributes).

A *call-party state* is a packaging of the fuller expression of a party's status (which may be multifaceted)—a compound state made up of the following groups:

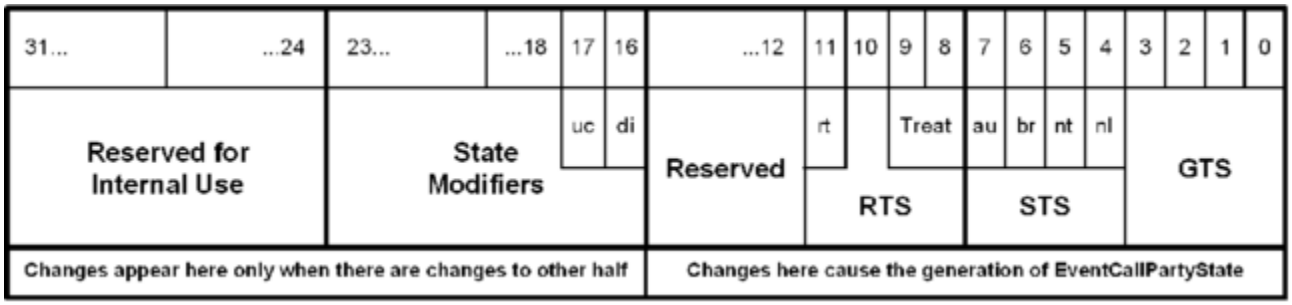
- Generic Telephony State (GTS);
- Supplementary Telephony State (STS);
- Routing/Treatment State (RTS).
- State modifiers

Each of these groups has parts that are referred to as elementary states, or properties (the basic attribute of a call-party). Typically, a call-party state consists of just one of these groups. However, there are certain cases where a coexistence of multiple groups in one call-party state is the norm. These, for instance, should be familiar:

- Routing with Queueing involves GTS and RTS.
- Service Observing involves both GTS and STS.

> **Important**
>
> Although not all combinations of elementary states make sense, there is no restriction on them.

A state is packed into one integer according to the structure shown below:



Call-Party Structure

Availability of State Information

The Genesys Framework assumes a particular call-party state for each call-party in the enterprise and supports the generic party-state set common to those T-Servers that support it. Not all party states or party-state transitions are mandatory for each T-Server. A given vendor-specific CTI may not provide a specific call-party state. Or, if a given state is available with the CTI, it may lack some relevant information required for T-Server to represent that particular state on the Genesys side of the software, or permit Genesys to perform a party-state transition. Thus the implementation of call-party states in T-Server is only complete to the extent that information is available from the switch. See `PartyState` in your API reference for details and state numeric values.

Generic Telephony State

The *Generic Telephony State* part of a call-party state comprises the most important of the sub-party states.

> ### Important
> Previously, the Genesys model did not include the states `Initiated` and `Failed` as marking participation in a call. As a result, there were no special events in DN-based event reporting to indicate a transition between those states and `NULL`. Some T-Servers used device-related `EventOffHook/OnHook` with at tribute `ConnID` to indicate a party had been added or deleted, but this event is optional and cannot be used when there is another active call on the same device (otherwise the status of the device is reported incorrectly). It is not possible to reconstruct the `Initiated` and `Failed` states from DN-based reporting. Furthermore, for compatibility reasons, parties in these states are not included in the `EventPartyInfo` list, and related calls are not included in `EventRegistered` and `EventAddressInfo`.

The Generic Telephony State has the following possible properties:

Null and Null[2]—Represent intermediate states when party information exists in T-Server memory, but when there is no logical relation between a call and device.

Initiated—Indicates that a call has been initiated by a device. Any telephony device that is initiating a new call enters the `Initiated` state while it awaits the availability of switch resources or user input. That is, the device remains in the `Initiated` state from the moment it goes off-hook until `EventDialing` is sent.

Queued—Identifies that a call is queued or parked on a given device, and that it awaits the availability of some service (for example, ACD queue distribution) or of some device (for example, a phone line).

Alerting—Means that a call is alerting on a device, indicating an incoming call (for example, the phone is ringing), or that a call is in the process of being distributed to a destination (for example, is being processed by the telephony network).

Connected—Indicates that a given device has a voice connection with other participants.

> **Important**
>
> The `Dialing` state (which is implicitly used in the DN call model) does not exist in the unified party-state model. Dialing was an attempt to report the state of the other party on the call. Such other parties cease to be clear in complex scenarios where transfers and conferences confuse matters. However, `Dialing`, as a state modifier, is available for compatibility with traditional reporting.

Call Progress (CP) Detection—The originating device (either a queue or a route point) for predictive dialing is put in this state from the time a call is made (and `EventDialing` is sent) to the moment the call is classified, at which time it is either moved to the `Queued` state, or is released.

Held—A device has temporarily suspended its connection to other call participants.

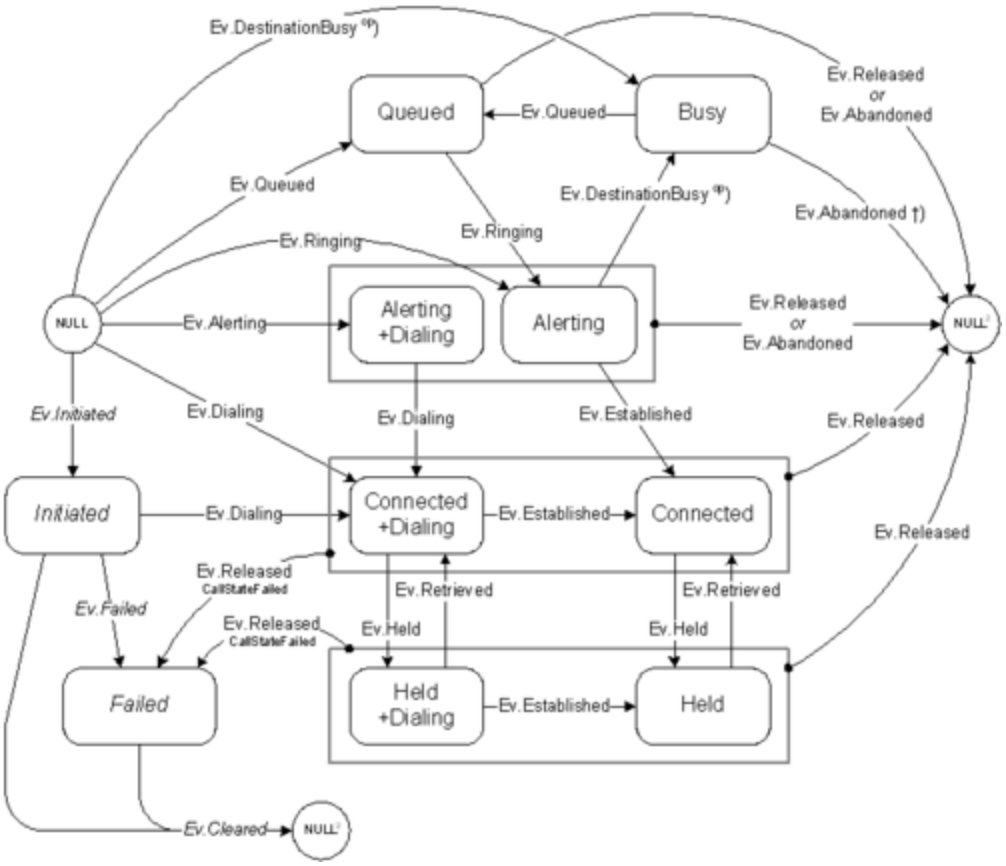
Busy—A call cannot reach the intended device, which is busy.

Failed—Indicates that a call originating on a given device has not succeeded (either the dialed number is wrong, or the switch was not able to allocate the trunk). This state is used instead of `Busy` when a destination party was never created for the call. `Failed` also applies to a party whose call has been disconnected, but who remains off hook.

Generic Telephony State Diagram for Regular DNs

The Generic Telephony State for Regular DNs diagram indicates the event flow that leads to the various sub states that comprise the Generic Telephony State. This diagram applies to all regular DNs. That is, all types except ACD queues and route points.

> **Important**
>
> Although they appear in the Generic Telephony State for Regular DNs diagram, events `Alerting`, `Initiated`, `Failed`, and `Cleared` do not currently exist. These names are reserved for future use.



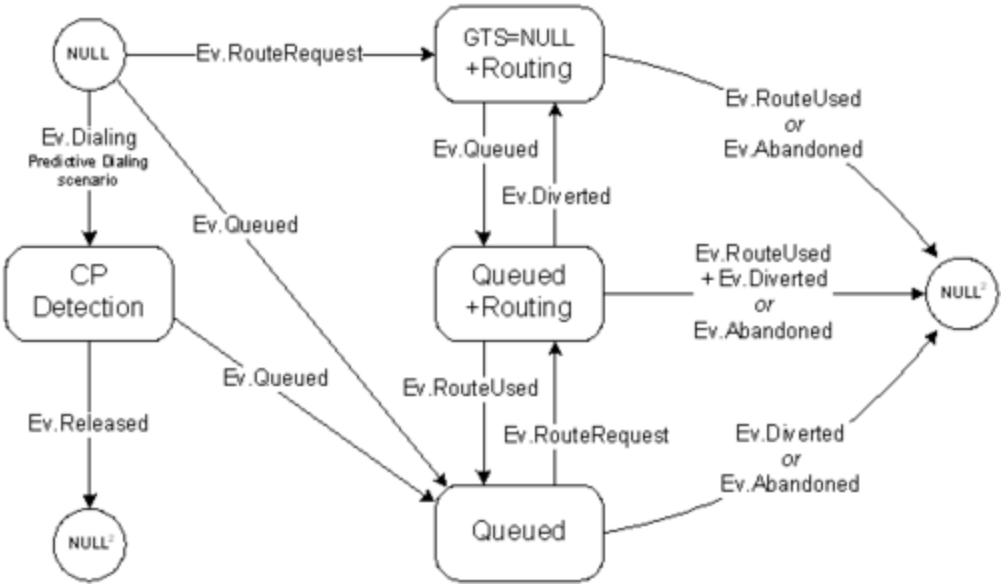
Generic Telephony State for Regular DNs

op Note that in this case, with a DN-based call model, `EventDestinationBusy` is reported for the opposite party.

† `EventAbandoned` is only sent if there is an `EventRinging` for that party. This occurs only with external parties.

Generic Telephony State Diagram for ACD Queues and Route Points

The Generic Telephony State for ACD Queues and Route Points diagram indicates the event flow that leads to the various sub states that comprise the Generic Telephony State. This diagram applies to ACD queues and route points.



Generic Telephony State for ACD Queues and Route Points

Supplementary State

The *Supplementary State* refers to combinations of the following party properties. Not all combinations of these properties make logical sense, but the general model does not put any restrictions on them. Furthermore, there is no transitional model for these properties (any state can change to any other).

NoListen—A party cannot hear other participants on the call.

NoTalk—A party is muted, and other participants on the call do not hear that party.

Audit—A party is attached to a call as `Service Observer` or `Service Assistant`.

Bridged—A party is attached to the call with the Bridged Call Appearance feature.

State Modifiers

State Modifiers further nuance the relationships between events and call states by allowing for intermediate sub states. These sub states provide additional information to the system, but may otherwise be unnecessary in the unified call-party state model.

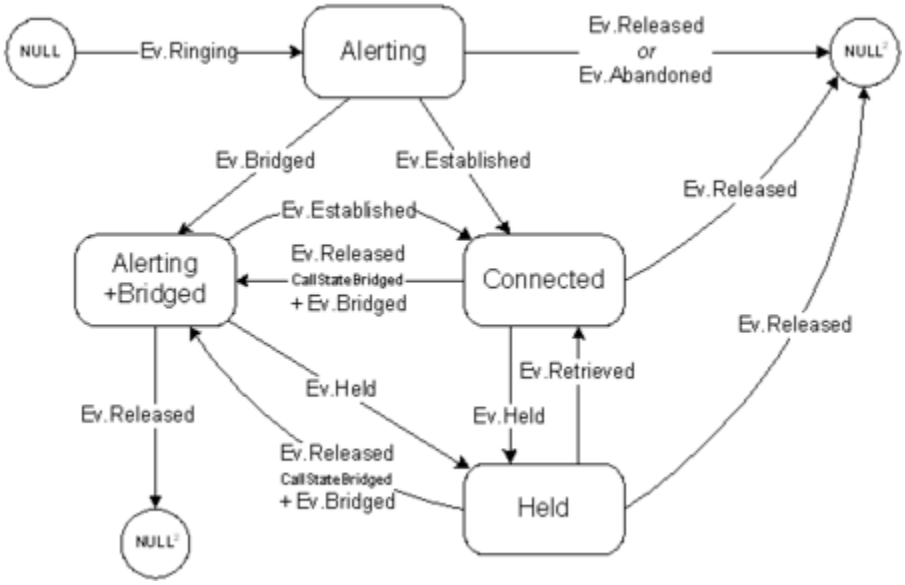
Dialing Modifier

The unified call-party state model uses `Dialing`, although not actually a party state, as a state modifier. This allows the system to handle the following scenarios, also illustrated in Modifier Dialing:

- Reporting applications may need to measure dialing time, which is defined as the interval between the moment a party is connected to a call and the moment when the voice connection with the other participant is established for the first time. It is possible to reconstruct the time spent in a dialing state without this modifier, but, since T-Server does not provide historical data, this calculation needs to be done when the full history of the call is available (by factoring in the states of other parties). The `Dialing` modifier provides clients access to this information during the call.
- T-Server needs an indication of whether `EventEstablished` has been distributed for a given party. In a DN-based call model this event is sent only when a connection is first established. (See Modifier Dialing for an instance of why a subsequent EventEstablished would be useful.) While it is possible to have this indication stored in device-specific states, the `Dialing` modifier allows for common functionality across media devices.

Uncertain Modifier

This modifier is set when the party information is not reliable. See `Reliability` in your API reference for details.



Modifier Dialing

Routing/Treatment State

The *Routing/Treatment State* consists of the following properties:

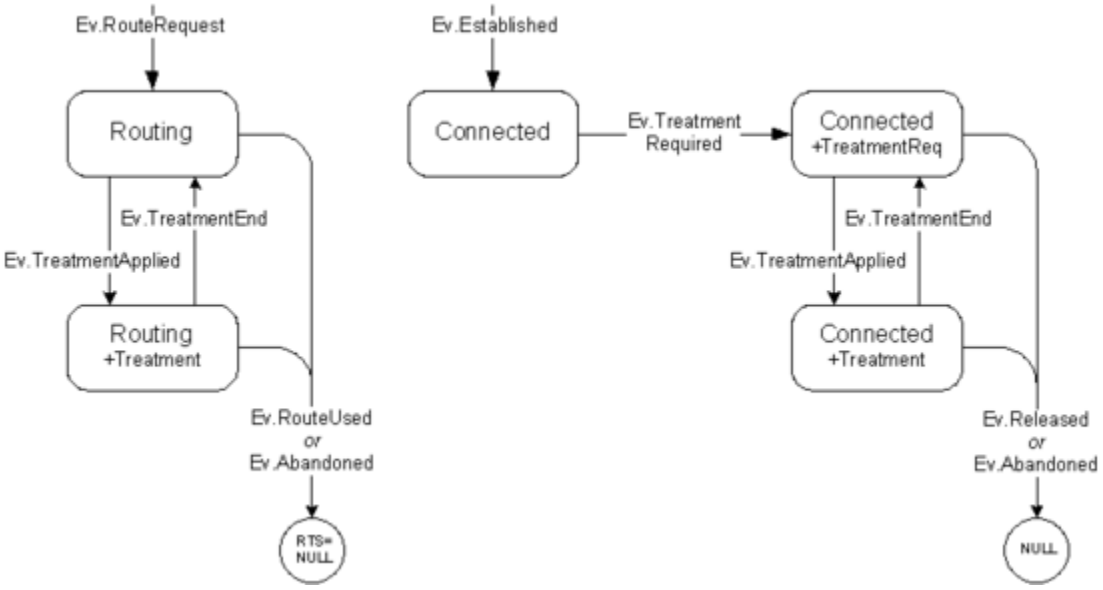
TreatmentReq—The switch is waiting for a treatment to be applied to the call.

Treatment—A treatment has been applied while the call is located on a given device.

Routing—The switch is waiting for routing instructions.

Routing/Treatment State Diagram

The following figure indicates the event flows that lead to the various states comprising the `Routing` (left portion of the diagram) and `Treatment` (right portion of the diagram) states. The `Routing` portion pertains to route points; the `Treatment` portion pertains directly to monitored IVR ports.



Routing (left portion) and Treatment States (right portion)