Genesys Info Mart Deployment Guide

Propagation Rules

5/10/2025

Propagation Rules

When you configure the user data mapping for Genesys Info Mart, also specify the propagation rule that Genesys Info Mart will use to determine what value to store if more than one value is extracted for the same key in the same interaction.

Contents

For user-data keys that might have multiple values over the life of an interaction, the propagation rules enable you to specify which KVP value will be stored for a particular INTERACTION_RESOURCE_FACT (IRF) or MEDIATION_SEGMENT_FACT (MSF) record, based on who changed (added, updated, or deleted) the KVP value or when it was changed.

Valid values for the propagation rules are the following:

- CALL — Genesys Info Mart stores the latest KVP value that is associated with the interaction when the interaction leaves the resource that is the subject of the IRF or MSF record, regardless of who changed the KVP value.

  The CALL propagation rule is suitable for most business requirements for most KVPs.

- PARTY — Genesys Info Mart stores the latest KVP value as changed by that party to the interaction, regardless of when it was changed.

  Use the PARTY propagation rule to capture KVP values that are set after the interaction leaves the agent (for example, during after call work [ACW]) or for user data that should be associated only with the subject of the IRF or MSF record and not propagated to other resources.

> **Important**
>
> - PARTY is the only propagation rule that enables you to capture KVP values that are set after the interaction leaves the handling resource that is the subject of the IRF record.
> - Do not use the PARTY rule for user data that is associated with virtual queues for multimedia interactions. User-data transformation for the PARTY rule relies on party information from the target IRF that is not available at mediation time.
> - The PARTY rule does not work for user data in MSFs for voice or multimedia interactions that are cleared or abandoned in a virtual queue. In these cases, Genesys Info Mart cannot find user data for the MSF, and it will appear as if no user data was changed. Genesys Info Mart will store the default value for the KVP, if defined.

- IRF — For a KVP value that was changed during the timespan of the IRF (specifically, between the time that the interaction started mediation to the handling resource and the time that the interaction leaves the handling resource), Genesys Info Mart stores the latest KVP value that is associated with the interaction, regardless of who changed the KVP value.

  The initial state of user data for the applicable IRF record is empty.

  For user data in MSFs, the IRF to which the MSF belongs provides the context in which the KVP value is set. For example, if MSF1 represents the mediation that occurs before the interaction is distributed to the resource that is the subject of IRF1, Genesys Info Mart will store the latest change, if any, that occurred during the MSF1 mediation interval, which is calculated from the mediation start time of IRF1 and the end time that is reported in MSF1.

- IRF_FIRST_UPDATE — For a KVP value that was changed during the extended timespan of the IRF for a handling resource, Genesys Info Mart stores the first update to the KVP value that is associated with the IRF, regardless of who changed the KVP value. The extended timespan starts from the time that the interaction started mediation to any handling resource (in other words, not necessarily to the resource that is the subject of the IRF) up to the time that the interaction leaves the handling resource in question. In a scenario with call redirection, the timespan of the IRF for the handling resource to which the interaction is eventually routed includes the durations of all preceding IRFs that have a technical result of Redirected/RoutedOnNoAnswer or Redirected/Unspecified.

  For user data in MSFs, the IRF to which the MSF belongs provides the context in which the KVP value is set. For example, if MSF*n* represents the mediation that occurs immediately before the interaction is distributed to IRF*n,* Genesys Info Mart will store the first change that was made during the extended timespan of IRF*n,* up to and including MSF*n.*

- IRF_INITIAL — Genesys Info Mart stores the KVP value that is associated with the interaction when the interaction enters the

resource that is the subject of the IRF or MSF record.

Use the IRF_INITIAL propagation rule to capture KVP values at the time that the call first rings at Agent1. For example, when an agent answers a call and updates a KVP value during the call, the value stored in the agent's IRF record is the initial value that the agent received with the call.

The IRF_INITIAL propagation rule was introduced in release 8.5.002.

- IRF_ROUTE — Genesys Info Mart stores the final KVP value that is present during mediation of an interaction:
  - If the call is delivered to a handling resource, Genesys Info Mart stores the KVP value that is associated with the interaction when the interaction enters the handling resource.
  - If the call is abandoned during mediation, Genesys Info Mart stores the last value that was in effect during mediation.

  For example, a call enters a routing point and the KVP value is initially set to Value1, and later updated to Value2. The caller either hangs up during mediation, or the call is routed to an agent, who updates the value to Value3. In either case, the IRF_ROUTE propagation rule records Value2.

  The IRF_ROUTE propagation rule was introduced in release 8.5.006.

Propagation Rule Examples

To illustrate the effect of the propagation rules, the following two tables provide the reporting results for the various propagation rules in typical sample scenarios:

- **Reporting Results for Propagation Rules — Sample Scenarios for IRF** provides examples for user data in IRF records.

  [+] Show table

Reporting Results for Propagation Rules — Sample Scenarios for IRF

Scenario*	User Data Result for Associated IRF Record, by Propagation Rule					
CALL	PARTY	IRF	IRF_FIRST_UPDATE	IRF_INITIAL	IRF_ROUTE	
1. A strategy attaches a KVP with Value0 and routes the call to Agent1, who updates the KVP to Value1 and then transfers	Agent1: Value1 Agent2: Value1	Agent1: Value1 Agent2: Empty	Agent1: Value1 Agent2: Empty	Agent1: Value0 Agent2: Empty	Agent1: Value0 Agent2: Value1	Agent1: Value0 Agent2: Value1

Scenario*	User Data Result for Associated IRF Record, by Propagation Rule					
the call to Agent2.						
2. Same as Scenario 1, except that Agent2 subsequently updates the KVP to Value2 after the call is released.	Agent1: Value1 Agent2: Value1	Agent1: Value1 Agent2: Value2	Agent1: Value1 Agent2: Empty	Agent1: Value0 Agent2: Empty	Agent1: Value0 Agent2: Value1	Agent1: Value0 Agent2: Value1
3. Same as Scenario 1, except that Agent 2 subsequently transfers the call to a routing point where the routing strategy updates the KVP to Value2 prior to the call being abandoned in the routing point.	Agent1: Value1 Agent2: Value1 RP: Value2	Agent1: Value1 Agent2: Empty RP: Value2	Agent1: Value1 Agent2: Empty RP: Value2	Agent1: Value1 Agent2: Empty RP: Value2	Agent1: Value0 Agent2: Value1 RP: Value1	Agent1: Value0 Agent2: Value1 RP: Value2
4. Strategy1 attaches a KVP with	Agent1: Value2 Agent1 (Initiated Consult):	Agent1: Value1 Agent1 (Initiated Consult):	Agent1: Value2 Agent1 (Initiated Consult):	Agent1: Value0 Agent1 (Initiated Consult):	Agent1: Value0 Agent1 (Initiated Consult):	Agent1: Value0 Agent1 (Initiated Consult):

Scenario*	User Data Result for Associated IRF Record, by Propagation Rule					
Value0 and routes the call to Agent1, who updates the KVP to Value1 and then initiates a two-step conference to Agent2, using Strategy2, which updates the KVP to Value2.	Value2 Agent2: Value2	Empty Agent2: Empty	Value2 Agent2: Value2	Value2 Agent2: Value2	Value1 Agent2: Value2	Value1 Agent2: Value2
5. A route-on-no-answer (RONA) variant of Scenario 3: Strategy1 attaches a KVP with Value0 and attempts to route the call to Agent0. When Agent0 does not answer, Strategy1 routes the call	Agent0: Value0 Agent1: Value2 Agent1 (Initiated Consult): Value2 Agent2: Value2	Agent0: Empty Agent1: Value1 Agent1 (Initiated Consult): Empty Agent2: Empty	Agent0: Value0 Agent1: Value1 Agent1 (Initiated Consult): Value2 Agent2: Value2	Agent0: Value0 Agent1: Value0 Agent1 (Initiated Consult): Value2 Agent2: Value2	Agent0: Value0 Agent1: Value0 Agent1 (Initiated Consult): Value1 Agent2: Value2	Agent0: Value0 Agent1: Value0 Agent1 (Initiated Consult): Value1 Agent2: Value2

Scenario*	User Data Result for Associated IRF Record, by Propagation Rule					
to Agent1, who updates the KVP to `Value1` and then initiates a two-step conference to Agent2, using Strategy2, which updates the KVP to `Value2`.						

* T-Server settings: **merged-user-data**=`merged-over-main`, **consult-user-data**=`inherited`

- **Reporting Results for Propagation Rules — Sample Scenarios for MSF** provides examples for user data in MSF records, provided that Genesys Info Mart has been configured to record the association.

[+] Show table

Reporting Results for Propagation Rules — Sample Scenarios for MSF

Scenario*	User Data Result for Associated MSF Record, by Propagation Rule					
CALL	**PARTY**	**IRF**	**IRF_FIRST_UPDATE**	**IRF_INITIAL**	**IRF_ROUTE**	
Note: The MSFs for Queue1 and Queue2 happen before the resulting IRFs for Agent1 and Agent2, respectively.						
1. A strategy attaches a KVP with `Value0` and places a call in Queue1. The call stays in the queue for a few	Queue1: `Value0` Queue2: `Value1`	Queue1: Empty Queue2: Empty	Queue1: `Value0` Queue2: Empty	Queue1: `Value0` Queue2: Empty	Queue1: `Value0` Queue2: `Value1`	Queue1: `Value0` Queue2: `Value1`

Scenario*	User Data Result for Associated MSF Record, by Propagation Rule					
days, before it is distributed to Agent1, who updates the KVP to `Value1` and then transfers the call to Agent2 via Queue2.						
2. Same as Scenario 1, except that a strategy updates the KVP to `Value2` when it places the interaction in Queue2.	Queue1: `Value0` Queue2: `Value2`	Queue1: Empty Queue2: Empty	Queue1: `Value0` Queue2: `Value2`	Queue1: `Value0` Queue2: `Value2`	Queue1: `Value0` Queue2: `Value2`	Queue1: `Value0` Queue2: `Value2`

* T-Server settings: **merged-user-data**=`merged-over-main`, **consult-user-data**=`inherited`