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.

Task Attributes and Interaction Properties

Working with Task Attributes and Interaction Properties

12/19/2025

Contents

Working with Task Attributes and Interaction Properties

These topics describe the task and attribute properties that are supported in iWD 8.5.

Task Attributes

Most of the iWD task attributes can be set when a task is created or updated through a Capture Point, although some, such as interaction ID, are set by iWD components or by Interaction Server. Most of the task attributes are displayed in the **Attributes** tab when a task is selected in the Global Task List. These attributes are maintained as pieces of attached data of the interaction, as it is stored in the Interaction Server's interactions database table. Some of the attributes are stored in independent columns in that database table, while others are stored in a binary (BLOB) format in a column in the interactions table called `flexible_properties`.

There are many reasons to update or access the data stored in the iWD task attributes, including:

- Setting the value of one or more task attributes as part of an iWD message such as `CreateTask`, when working with a capture point.
- Reading or updating task attributes in business rules.
- Using the data in the **Condition**, **Order**, and **Segmentation** tabs of Views in Genesys Business Processes.

> **Important**
>
> You cannot use properties with a `Timestamp` data type on the **Segmentation** tab.

- Reading or updating the data contained in task attributes within a routing strategy.
- Making the data available to an agent or knowledge worker desktop application, either to display to the agent or to facilitate a screen pop.
- Filtering the display of the Global Task List.

[+] CORE ATTRIBUTES

Core attributes describe the fundamentals of a task. These attributes are used in assembling tasks in the Global Task List, based on the business value and priority that are defined within iWD. Core attributes are either set automatically by iWD, or provided by the source system (through the Capture Point interface). The following are some iWD core task attributes:

- `activationDateTime`

- `assignedDateTime`
- `assignedToUser`
- `businessValue`
- `captureId`
- `category`
- `completedDateTime`
- `dueDateTime`
- `expirationDateTime`
- `heldDateTime`
- `interactionid`
- `mediaChannel`
- `priority`
- `queue`
- `queueTarget`
- `queueType`

[+] EXTENDED ATTRIBUTES

Extended attributes provide additional context about a task, enabling you to tailor the service-level agreement (SLA) rules for managing tasks on the Global Task List. They can also aid in customizing current-day and historical reporting. For example, use of several capture dates allows an organization to measure performance against the date and time at which an order or loan application was received by the source system or was submitted by the customer via a web form. The following are some extended task attributes:

- `customerId`
- `customerSegment`
- `productSubtype`
- `productType`
- `requestedAgent`
- `requestedSkill`
- `resultCode`
- `sourceCreatedDateTime`
- `sourceDueDateTime`
- `sourceFirstCreatedDateTime`
- `sourceProcessSubtype`
- `sourceProcessType`

- sourceTenant

Task Attribute Mapping

Task Attribute Mapping

Task Attribute	Column Name in Interactions Table	Type	Interaction Attached Data Key	Label in Global Task List
Core Attributes				
Age	received_at, moved_to_queue_at, queue, completed_at	String	ReceivedAt, MovedToQueueAt, Queue, CompletedAt	Age
Queue	queue	String	Queue	Queue
QueueType	workbin, agent_id, agent_group_id, place_id, place_group_id	String	Workbin, WorkbinAgentId, WorkbinAgentGroupId, WorkbinPlaceId, WorkbinPlaceGroupId	Queue Type
QueueTarget	agent_id, agent_group_id, place_id, place_group_id	String	WorkbinAgentId, WorkbinAgentGroupId, WorkbinPlaceId, WorkbinPlaceGroupId	Queue Target
MovedtoQueueD/T	moved_to_queue_at	Timestamp	MovedToQueueAt	Moved to Queue D/T
BrokerId	Id	String	InteractionId	ID
Status—See note below table.	queue	String	Queue	Status
mediaType	media_type	String	MediaType	Media Type
TenantId	IWD_tenantId	String	IWD_tenantId	Tenant
businessCalendarId	in <flexible_properties>	String	IWD_businessCalendarId	Business Calendar ID

Task Attribute	Column Name in Interactions Table	Type	Interaction Attached Data Key	Label in Global Task List
DepartmentId	IWD_departmentId	String	IWD_departmentId	Department (name is shown instead of ID)
ProcessId	IWD_processId	String	IWD_processId	Process (name is shown instead of ID)
Channel	IWD_channel	String	IWD_channel	Channel
Category	IWD_category	String	IWD_Category	Category
CapturePointID	IWD_capturePointId	String	IWD_capturePointId	Capture Point (value is shown instead of ID)
CaptureId	external_id	String	ExternalId	Capture ID
CreatedDateTime	received_at	Timestamp	ReceivedAt	Created D/T
DistributionPointId	IWD_distributionPointId	String	IWD_distributionPointId	Distribution Point (value is shown instead of ID)
ActivationDateTime	IWD_activationDateTime	Timestamp		Activation D/T
BusinessValue	IWD_businessValue	Integer	IWD_businessValue	Business Value
DueDateTime	IWD_dueDateTime	Timestamp	IWD_dueDateTime	Task Due D/T
Priority	priority	Integer	Priority	Priority
ReprioritizeDateTime	IWD_reprioritizeDateTime	Timestamp	IWD_reprioritizeDateTime	Reprioritization D/T
AssignedToUser	assigned_to	String	RTargetAgentSelected	Assigned To
AssignedDateTime	assigned_at	Timestamp	AssignedAt	Assigned D/T
-	completed_at	Timestamp	CompletedAt	Completed D/T
ExpirationDateTime	IWD_expirationDateTime	Timestamp	IWD_expirationDateTime	Expiration D/T
-	IWD_solutionId	String	IWD_solutionId	-

Task Attribute	Column Name in Interactions Table	Type	Interaction Attached Data Key	Label in Global Task List
Extended Attributes				
CustomerSegment	IWD_ext_customerSegment	String	IWD_ext_customerSegment	Customer Segment
CustomerId	IWD_ext_customerId	String	IWD_ext_customerId	Customer ID
ProductType	IWD_ext_productType	String	IWD_ext_productType	Product
ProductSubtype	IWD_ext_sourceProductSubtype	String	IWD_ext_productSubtype	Subproduct
RequestedAgentGroup	IWD_ext_requestedAgentGroup	String	IWD_ext_requestedAgentGroup	Requested Agent Group
RequestedPlaceGroup	IWD_ext_requestedPlaceGroup	String	IWD_ext_requestedPlaceGroup	Requested Place Group
SourceTenant	IWD_ext_sourceTenant	String	IWD_ext_sourceTenant	TOS Tenant
SourceProcessType	IWD_ext_sourceProcessType	String	IWD_ext_sourceProcessType	TOS Process
SourceProcessSubtype	IWD_ext_sourceProcessSubtype	String	IWD_ext_sourceProcessSubtype	TOS Subprocess
SourceFirstCreatedDateTime	IWD_ext_sourceFirstCreatedDT	Timestamp	IWD_ext_sourceFirstCreatedDateTime	TOS First Created D/T
SourceCreatedDateTime	IWD_ext_sourceCreatedDateTime	Timestamp	IWD_ext_sourceCreatedDateTime	TOS Created D/T
SourceDueDateTime	IWD_ext_sourceDueDateTime	Timestamp	IWD_ext_sourceDueDateTime	TOS Due D/T
ResultCode	IWD_ext_resultCode	String	IWD_ext_resultCode	Result Code
RequestedAgent	IWD_ext_requestedAgent	String	IWD_ext_requestedAgent	Requested Agent
RequestedSkill	IWD_ext_requestedSkill	String	IWD_ext_requestedSkill	Requested Skill
Custom Attributes				
myCustomAttribute	in <flexible_properties>	String	myCustomAttribute	myCustomAttribute

Task Attribute	Column Name in Interactions Table	Type	Interaction Attached Data Key	Label in Global Task List
`ArchiveDestination` Archive Destination has been maintained from iWD 8.0 and 8.1.0 releases. However, due to changes in task archiving, it is no longer necessary to populate this attribute through business rules.	`in <flexible_properties>`	String	`ArchiveDestination`	Archive Destination
`ESP_Result`	`in <flexible_properties>`	String	`ESP_Result`	ESP_Result
`ESP_Error`	`in <flexible_properties>`	String	`ESP_Error`	ESP_Error

Important

The value of Status does not correlate directly to the contents of the queue column in the interactions table. It is dynamically calculated, taking into account information such as the queue and whether the task is held or not. Because the contents of the Status column are dynamically calculated, rather than being read from a database table, the Global Task List cannot be sorted by the Status column. You should use filters instead, if you are interested in focusing in on the contents of the list by this criterion.

Interaction Custom Properties

If you want to use the value of a task attribute in the `Condition`, `Order`, and `Segmentation` tabs of Views in Genesys Business Processes, or if you want to filter or sort the display of the Global Task List by using a task attribute, that task attribute must be represented in an independent column in the Interaction Server's `interactions` database table. If that task attribute is inside the binary data in the `flexible_properties` column, you must create an Interaction Custom Property that corresponds to that attribute. The data type of the property can be a timestamp, string, or number.

Important

Properties with the Timestamp data type cannot be used on the `Segmentation` tab of

> Views in a Genesys Business Process.

Configuring a custom interaction property

1. Decide on an attached data key that will be the source of the content of the custom property.
2. Create a new field directly in the `interactions` database.
3. Create a new Business Attribute:

   Name = `InteractionCustomProperties`

   Display name = `Interaction Custom Properties`

   Type = `Custom`

   If such an attribute already exists go to the next step.
4. Expand `Interaction Custom Properties` and open its `Attribute values.`
5. Give it an `Attribute Value`, with a name exactly matching the attached data key name that you decided on in Step 1. The matching is case sensitive (you can create a separate display name).
6. In your new attribute value, go to the `Annex` tab and create a section called `translation.`
7. In the new `translation` section, create an option called `translate-to,` with its value duplicating the name of the new field you created in Step 2.
8. If required, configure user-friendly labels for any custom attribute, that will appear in the Global Task List's `Attributes` tab, in the list of `Advanced Filters,` and the list of attributes that are used when you are building custom filters. See the **Configuring Custom Attributes** tab.

> ### Important
> If you specify a custom field as not null, you must ensure that you provide some data to that field upon creation of a task. If no data is provided, the request will fail because Interaction Server sends NULL for empty fields, and that will be rejected by the DBMS.

> ### Important
> There are specific columns in the `interactions` table that you should not change. Please refer to Chapter 8, “Interaction Properties”, in the eServices 8.1 User’s Guide.

Configuring Custom Attributes

When you capture a task from a source system and that task has custom attributes in it, you will need to configure the system properly so that each custom attribute is recognized. Several steps are required to ensure the custom attribute:

- Appears on the Global Task List with a user-friendly label and can be used in advanced and custom filters.
- Can be properly populated in iWD Data Mart.
- Can be used in the `Condition`, `Order`, and `Segmentation` tabs of `Views` in Genesys Business Processes.

Process Summary Table

Objective	Related Procedures and Actions
Add database columns and Interaction Custom Property for Custom Attributes.	1. Add a unique column in the Interaction Server database interactions table to store the value of this custom attribute.
	2. Create a new Business Attribute to correspond to the custom attribute and map it to the new database column you added in the Interaction Server database interactions table.
	3. Add a unique column to two tables in the Interaction Server Event Log database to store the value of this custom attribute.
	4. Add two new options on the Interaction Server Event Log Database Access Point application to refer to the newly-added database columns in the Interaction Server Event Log database.
Edit the resource file on the application server.	5. Edit this file to indicate how you want the custom attribute to be labeled on the Global Task List. See the Editing the iWD Manager’s Resource File tab on this page.

Adding new database column(s) and Interaction Custom Property for each Custom Attribute

To add new database columns and interaction custom properties to custom attributes, do the following:

1. Decide on the name of the interaction user data key that will store the value of your custom attribute.

This is the value you will use in the `CreateTask`message when you create a new task from an iWD capture point (see description of the `CreateTask` message on the **Create Task** tab).

> ## Important
>
> If your custom attribute is going to be used to store a timestamp, ensure the name of the interaction user data key ends with `DateTime`, (for example, `CustomDateTime`). Then, the custom attributes that appear on the `Attributes` tab of the Global Task List will be properly formatted as date and time. For example, `December 31, 2012 9:30 PM`, instead of `2012-12-31T21:30:00Z`.

2. In your database server’s Administration Console, add a new column to the Interaction Server database interactions table.

   This column can be a timestamp, string, or number. The exact data types will differ depending on the type of database server you are using. The name of this column does not necessarily need to match the interaction user data key that is storing the custom attribute.

> ## Important
>
> If your custom attribute is going to be used to store a timestamp, ensure this column name ends with `DateTime`, (for example, `CustomDateTime`). Then, when you create a Global Task List filter with this custom attribute or use it in an advanced filter, the user will see a calendar control to pick the date.
>
> Properties with the `Timestamp` data type cannot be used on the `Segmentation` tab of `Views` in a Genesys Business Process.
>
> If you specify a custom database field as not null, you must ensure that you provide some data to that field upon creation of a task. If no data is provided, the request will fail because Interaction Server sends `NULL` for empty fields, which will be rejected by the DBMS.

3. In Genesys Administrator or Configuration Manager, create a new Business Attribute under the tenant you are working with for this iWD Solution (if an attribute already exists go to the next step):
   a. Create a new Business Attribute with the following properties:
      - Name: `InteractionCustomProperties`
      - Display Name: `Interaction Custom Properties`
      - Type: `Custom`

   b. Expand `Interaction Custom Properties` and open `Attribute values`. The name of the `Attribute value` must match exactly the interaction user data key name that you used in Step 1. The matching of names is case-sensitive. (You can create a separate display name.)

   c. In the new `Attribute value`, go to the `Annex` tab and create a section named `translation`.

   d. In the `translation` section, create an option named `translate-to`, with a value name that matches the name of the database field you created in Step 2.

> ### Important
>
> Steps 4 and 5 will affect the correct population of the custom task attribute in iWD Data Mart and ensure that the Global Task List's `Historytab` is correctly populated. If you complete these steps, the `History` tab will display an event whenever the value of the custom attribute is updated.

5. In your database server's Administration Console, add a new column to the Interaction Server Event Log database, in both the `rpt_esp` and `rpt_interaction` tables.

   Using the same data type for this column as you did in Step 2, add a new column to the Interaction Server database interactions table.

6. Create two new options on the Interaction Server Event Log Database Access Point application, as follows:

   a. Using Genesys Administrator or Configuration Manager, open the Interaction Server Event Log Database Access Point application object.

   b. On the `Application Options` tab, in the `esp-custom-data` section, add a new option with a name that matches the attached data key from Step 1, and using a value that matches the new database column that you added in Step 4, add a new column to the Interaction Server Event Log database, in both the `rpt_esp` and `rpt_interaction` tables.

   c. In the `itx-custom-data` section, add a new option with a name that matches the attached data key from Step 1, and using a value that matches the new database column added in Step 4, add a new column to the Interaction Server Event Log database, in both the `rpt_esp` and `rpt_interaction` tables.

4. Restart Interaction Server.

Editing the iWD Manager's Resource File

By default, the on-screen labels you will see on the Global Task List's Attributes tab of the Task Details panel for all custom task attributes will be the interaction user data keys. These labels may not be very user-friendly. For example they might include multiple words concatenated, or they might use underscore characters or have odd capitalization.

To configure user-friendly labels for custom task attributes by adding into a resources file, the labels that you want to see and that will be used by iWD Manager.

1. Stop your application server.
2. Inside the `iwd_manager` application folder, which will be in the `webapps` folder of your application server, navigate to the `.../WEB-INF/classes/evo/cmc/ui/resources` directory, or if you are using a localized version of iWD Manager, navigate to the `.../WEB-INF/classes/lang/evo/cmc/ui/resources` directory.
3. Create a new text file called `resources_custom.properties` (if it does not already exist).
4. In the `resources_custom.properties` file, add a line for each custom attribute for which you want to define an on-screen label, by using the following format:

   ```
   TASK_attribute_key=Attribute Label
   ```

Where:

- `attribute_key` matches the interaction user data key of the custom attribute.
- `Attribute Label` is the user-friendly string you want to display on the Global Task List. For example, if your custom attribute's user data key is `MyAttribute`, but you want it to display as `My Custom Attribute`

on the Global Task List, your entry would look like this: `TASK_MyAttribute=My Custom Attribute`.

> ## Important
>
> This label will be used in the following places in the Global Task List:
>
> - As the label for this attribute on the `Attributes` tab of the `Task Details` panel.
> - In the `Select` columns to add drop-down list on the `Filters` page.
> - As the column header if you add this custom attribute as a column in the task table when you are creating a custom filter.

- In the same `resources_custom.properties` file that was used in the **Adding new database column(s) and Interaction Custom Property for each Custom Attribute procedure**, add a new line for each custom attribute, by using the following format:

  `FILTER_ATTR_attributedb_column_name=Attribute Label`

  Where:

  - `attributedb_column_name` matches the column name that you added to the interactions table.
  - `Attribute Label` is the user-friendly string you want to display in the Global Task List. For example, if the name of the database column is `my_attribute`, this line in the `resources_custom.properties` file would look like this: `FILTER_ATTR_my_attribute=My Custom Attribute`.

> ## Important
>
> This label will be used in the following places in the Global Task List:
>
> - In the `Advanced Filters` drop-down list.
> - In the `attributes` drop-down list that you use to add a custom attribute in a filter criterion, when you are creating or editing a Global Task List filter.

- Restart the `iwd_manager` application on your application server for the changes to become effective.

Important

If the user-friendly labels need to be localized, create a separate file for each supported locale. Use the following name pattern for the file: `resources_custom_xx[_YY].properties`, where `xx` and YYare language code and country code, respectively. Also, do not use Unicode or any language-specific encoding for the properties files. Use ISO-8859-1 encoding, which supports only a few Western languages. Encode resources in other languages by using escape sequences, such as the native2ascii program, which is a part of the Oracle JDK.