Data Tables

Section	Description
Feature Area	Architect / Orchestration Assets
Navigation	`Admin → Architect → Data Tables`
Alt Navigation	`Menu → Orchestration → Orchestration Assets → Data Tables`
Primary Function	Store configuration data locally so Architect flows can look it up dynamically during an interaction
Typical Use Cases	CRM-to-queue mapping, account routing, dynamic prompt selection, large lookup sets that exceed switch statement limits
Key Dependency	Architect flows use the **Data Table Lookup** action to retrieve values at runtime

Data tables allow you to store data locally so Architect can access it within an interaction. They are particularly useful for data sets larger than what a switch statement can handle, and for combining Genesys Cloud with third-party integrations — for example, mapping an account number returned by a CRM to a Genesys Cloud queue.

> **Important:** Data tables are intended for **configuration data only**. They must not contain personal information or any data that could identify a natural, living person.

Study Notes

Topic	Explanation
Data Table	A structured lookup table stored within Genesys Cloud, accessible by Architect flows
Reference Key	The **primary key** of the table — uniquely identifies each row. Required. Cannot be changed after a row is created.
Reference Key Label	A descriptive name for the reference key column (e.g., "Account Number"). Cannot include `/` or `\`

Topic	Explanation
Custom Fields	Additional columns beyond the reference key. Up to **50 fields per table**. Cannot be deleted after the table is saved.
API Field ID	Auto-populated from the field label — used when loading data via API. Cannot be changed after creation.
Data Table Lookup Action	The Architect action used inside a flow to query a data table by reference key and map results to flow variables
Division	Each data table is assigned to a division for access control
Import/Export	Tables support CSV import (append or replace) and export

---

Org Limits (Exam Critical)

Limit	Value
Maximum data tables per organization	**200**
Maximum rows per table	**5,000**
Maximum fields per table	**50**
Maximum characters in a reference key value	**256**
Maximum characters in a table name	**256**

> To request higher limits, contact **Genesys Cloud Customer Care**.

Permissions

Action	Permission Required
View data table	`Architect > Datatable > View`
Add data table	`Architect > Datatable > Add`
Edit data table	`Architect > Datatable > Edit`
Delete data table	`Architect > Datatable > Delete`
View data table row	`Architect > Datatable Row > View`

Action	Permission Required
Add data table row	`Architect > Datatable Row > Add`
Edit data table row	`Architect > Datatable Row > Edit`
Delete data table row	`Architect > Datatable Row > Delete`
All permissions shortcut	`Architect > Datatable > All` + `Architect > Datatable Row > All`

Navigation

Task	Navigation Path
Open Data Tables	`Admin → Architect → Data Tables`
Alt Navigation	`Menu → Orchestration → Orchestration Assets → Data Tables`
Create a table	Click **Add**
Edit a table	Hover over row → click **Edit**
Delete a table	Hover over row → click **Delete** (cannot delete if table is in use by a flow)
View table rows	Click the table name or hover → click **View**
Import data	Click **Manage Imports**
Export data	Click **Export Data**

Configuration Fields (UI Form Fields)

Data Tables List Page

UI Field	Description
Name	Table name — unique, max 256 characters, no duplicates
Reference Key Label	Describes the primary key purpose (e.g., "Account Number") — no `/` or `\`

UI Field	Description
Description	Optional — helpful context about the table
Division	Division the table belongs to
Export Data	Exports table rows to CSV
Manage Imports	Import rows via CSV (append or replace)
Delete	Delete selected tables (cannot delete a table used in a flow)
Refresh	Reload the table list
Add	Create a new data table
View (hover)	View table rows
Edit (hover)	Edit table structure or rows
Delete (hover)	Delete individual table

Create Data Table Form

UI Field	Description	Notes
Name	Unique table name	Max 256 characters; no duplicates
Division	Division assignment	Required
Description	Optional context	Free text
Reference Key Label	Name for the primary key column	Cannot include `/` or `\`
Add Field → Boolean	Checkbox field	Set "True by default" option
Add Field → Integer	Whole number field	Set default value
Add Field → Decimal	Decimal number field	Set default value
Add Field → String	Text field	Set default text value
API Field ID	Auto-populated from field label	Read-only — cannot be changed after creation
Save	Save table	Required before adding rows

> **Permanent limitations once saved:**
>
> - You **cannot delete a custom field** after saving the table

> - You **cannot change the API Field ID** of a custom field — only the label can be changed
> - You **cannot modify the reference key value** of an existing row

---

Add Table Row Form

UI Field	Description
Reference Key	Value for the primary key (e.g., the account number)
Custom Field values	One entry per configured custom field
Save & Close	Save row and return to table
Save & Create Another	Save row and immediately add another

Data Table Lookup Action (in Architect Flows)

Attribute	Details
Action Name	**Data Table Lookup**
Purpose	Query a data table using a reference key value and map results to flow variables
Required Permission	`Architect > Data Table > All`
Input	Reference Key value (can come from a flow variable, e.g., entered by caller)
Output	Custom field values mapped to flow variables
Failure Path	Flow continues via failure path if key is not found

Example use in a flow:

```
Caller enters Account Number
        ↓
Data Table Lookup action
```

Input: Account Number (reference key)
Table: CRM_Queue_Mapping
↓
Output: Queue Name → flow variable
↓
Transfer to Queue using variable

Import / Export

Feature	Description
Export	Downloads all table rows as a CSV file. Exported file retains original API field keys (not display labels).
Import	Upload a CSV to append rows or replace all existing rows
Manage Imports	View import history and any import errors

> When exporting, the CSV retains the **original API field keys**, not the display labels — relevant if labels were renamed after creation.

Dependencies

Component	Purpose
Architect Flows	Consume data tables via the Data Table Lookup action
Divisions	Control which users/flows can access a given table
CRM / External Systems	Common source of data loaded into tables
Flow Variables	Receive output values from Data Table Lookup

Platform Integration / Related Components

Component	Relationship
Inbound Call Flows	Most common flow type using Data Table Lookup
Inbound Message Flows	Can also use Data Table Lookup for digital routing
Data Actions	Alternative for real-time external API lookups (vs. static data in tables)
Decision Tables	Related feature under Rule-Based Decisions — logic-based conditional routing
Switch Action	Alternative for small, static lookup sets directly in the flow

Related Topics / Further Reading

Topic	Description
Data Table Lookup Action	Architect action that queries a data table at runtime
Import or Export Data Tables	Load data via CSV
Decision Tables	Rule-based routing decisions (separate feature under `Admin → Rule-Based Decisions`)
Architect Overview	Parent feature area
Flow Variables	Store and pass values within a flow

Implementation Checklist

Task	Status
Identify what data needs to be looked up in the flow	☐

Task	Status
Design the table schema (reference key + custom fields)	☐
Create the data table and define fields	☐
Populate rows (manually or via CSV import)	☐
Add the Data Table Lookup action to the flow	☐
Map output fields to flow variables	☐
Test the flow with known reference key values	☐
Handle the failure path (key not found)	☐

Implementation Guide

Step	Action
Step 1	Navigate to `Admin → Architect → Data Tables`
Step 2	Click **Add**
Step 3	Enter a unique **Name**, select a **Division**, add optional **Description**
Step 4	Enter a descriptive **Reference Key Label**
Step 5	Click **Save**
Step 6	Add **Custom Fields** (Boolean, Integer, Decimal, String)
Step 7	Save field configuration
Step 8	Add rows manually or import via **Manage Imports** (CSV)
Step 9	In Architect, add a **Data Table Lookup** action to your flow
Step 10	Configure the lookup: select table, set reference key input, map outputs to variables
Step 11	Handle the failure path for keys not found

Workflow

```
Admin Creates Data Table
        ↓
```

Fields Defined (Reference Key + Custom Fields)

↓

Rows Populated (Manual or CSV Import)

↓

Architect Flow Uses Data Table Lookup Action

↓

Caller Input (e.g., Account Number) Passed as Reference Key

↓

Matching Row Found → Custom Field Values Returned

↓

Values Stored in Flow Variables

↓

Flow Uses Variable (e.g., Queue Name) for Routing

Architecture Diagram

External CRM / System

↓

CSV Export

↓

Data Table (Genesys Cloud)

Reference Key: Account Number

Field 1: Queue Name

Field 2: VIP Flag

Field 3: Language Preference

↓

Architect Flow

Data Table Lookup Action

↓

Flow Variables → Routing Logic

Real Flow Scenarios

Scenario 1 — CRM-to-Queue Mapping

Caller enters Account Number via IVR

↓

Data Table Lookup: AccountNumber → DepartmentQueue

↓

Matched: "Billing" queue name returned

↓

Transfer to Billing Queue

Scenario 2 — VIP Routing

Caller enters Customer ID

↓

Data Table Lookup: CustomerID → VIPFlag, PreferredQueue

↓

VIPFlag = True → Transfer to Priority Queue

VIPFlag = False → Transfer to Standard Queue

Scenario 3 — Language Preference

Caller enters Account Number

↓

Data Table Lookup: AccountNumber → LanguagePreference

↓

Play prompt in preferred language

Usage Scenarios

Scenario	Description
CRM queue mapping	Map external account IDs to internal queues
VIP identification	Flag high-value customers for priority routing
Language preference routing	Route to language-appropriate queue
Dynamic prompt selection	Retrieve prompt names stored in table
Product-based routing	Look up department by product code
Configuration data storage	Store environment-specific values accessible to flows

Best Practices

Practice	Reason
Design your schema carefully before creating the table	Fields cannot be deleted after saving
Use descriptive Reference Key Labels	Makes the table purpose clear to other admins
Do not store personal data	Violates Genesys data use policy for data tables
Use Import/Export for bulk data management	Faster and more reliable than manual row entry
Always handle the Lookup failure path in the flow	Prevents unhandled routing failures when a key is not found
Keep table size within limits	Max 5,000 rows; contact Customer Care for higher limits
Use separate tables per business domain	Easier to maintain and audit
Validate imported CSV against table schema	API Field IDs in CSV must match table schema

Naming Convention

Resource	Example
Data Table	`CRM_AccountQueue_Mapping`
Data Table	`VIP_Customer_Flags`
Data Table	`Language_Preference_Lookup`
Reference Key Label	`Account Number` / `Customer ID`

Naming pattern:

<Source>_<Purpose>_<Type>

Security Considerations

Control	Description
Division Assignment	Controls which flows and users can access the table
Role-Based Permissions	Granular `Architect > Datatable` permissions per action
No PII	Data tables must not contain personal identifiable information
API Field ID immutability	Prevents accidental schema changes after deployment

Limitations / Constraints

Constraint	Value / Description
Max tables per org	200 (can request increase via Customer Care)
Max rows per table	5,000 (can request increase)
Max fields per table	50
Max reference key length	256 characters
Max table name length	256 characters
Cannot delete custom fields	Once saved, fields are permanent — create a new table if needed
Cannot change API Field ID	Only display labels can be changed after creation
Cannot change reference key value	Row reference key values are immutable once created
Cannot delete a table in use	Must remove flow references first
Reference key label restrictions	Cannot contain `/` or `\`

Troubleshooting

Issue	Cause	Resolution

Lookup returns no match	Reference key value not in table	Verify data is loaded; check for whitespace or case mismatch
Cannot delete table	Table is referenced in one or more flows	Remove Data Table Lookup references in flows first
Cannot delete a field	Fields are permanent after saving	Create a new table with the correct schema
Import fails	CSV column keys don't match API Field IDs	Export the table first to get the correct column headers
API Field ID unexpected	Label renamed after creation	API Field ID is fixed at creation; only label changed
Flow variable empty after lookup	Failure path taken — key not found	Check row exists; add default handling in failure path

---

Interview Cheat Sheet

Question	Answer
What is a data table in Genesys Cloud?	A locally stored lookup table that Architect flows can query during an interaction
What is the purpose of the Reference Key?	It is the primary key that uniquely identifies each row — used as the lookup input
What are the org limits?	200 tables / 5,000 rows per table / 50 fields per table
What can you NOT do after saving a data table?	Delete custom fields or change API Field IDs
What can you NOT do to a row's reference key?	Modify it — reference key values are immutable once created
What Architect action reads from a data table?	**Data Table Lookup**
Can data tables store personal information?	No — configuration data only
What navigation opens Data Tables?	`Admin → Architect → Data Tables` or `Menu → Orchestration → Orchestration Assets → Data Tables`
How do you load data in bulk?	CSV import via **Manage Imports**
Can you delete a table being used by a flow?	No — must remove flow references first

---

Key Takeaways

Topic	Summary

Data Tables	Local configuration data storage for Architect flows
Reference Key	Primary, immutable key for each row — required
Field Types	Boolean, Integer, Decimal, String — permanent after save
Limits	200 tables / 5,000 rows / 50 fields
Data Table Lookup Action	Retrieves values from a table at runtime using a reference key
No PII	Personal data must never be stored in data tables
Import/Export	CSV-based bulk data management
Failure Path	Always handle the case where a key is not found

Revision #1
Created 13 March 2026 07:04:01 by Cesar Gzz
Updated 13 March 2026 07:04:16 by Cesar Gzz