



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.

# Configuration Layer Objects Reference Guide

[CfgTableAccess](#)

---

## Contents

- [1 CfgTableAccess](#)
  - [1.1 Description](#)
  - [1.2 Filter Keys](#)
  - [1.3 Attributes](#)
  - [1.4 Comments](#)
  - [1.5 XML Representation](#)
  - [1.6 See Also](#)

# CfgTableAccess

## Description

*Table Access* objects describe database tables of a specified Format and explain how to access these tables through Database Access Points.

## Filter Keys

Filter Name	Type	Description
dbid	int	A unique identifier of a table access. If specified, configuration server will return information only about this table access.
tenant_dbid	int	A unique identifier of the tenant. If specified, Configuration server will return information only about the table accesses that belong to this tenant.
name	string	Name of a table access. Shall be specified as a character string. If specified, Configuration Server will return information only about the table access(es) with that name.
type	int	A type of table (See <a href="#">CfgTableType</a> ). If specified, Configuration Server will return information only about the table(s) of that type.
dbaccess_dbid	int	A unique identifier of the data base access point through which the table can be accessed. If specified, Configuration Server will return information only about the table access(es) which consists this data base access point.
format_dbid	int	A unique identifier of format of the table. If specified, Configuration Server will return information only about the table(s) with that format.
db_table_name	string	Name of the table. Shall be

Filter Name	Type	Description
		specified as a character string. If specified, Configuration Server will return information only about the table access(es) for that table.
state	int	Current state of the table access (See <a href="#">CfgObjectState</a> ). If specified, Configuration Server will return information only about the table access(es) that are currently in this state.

## Attributes

- **DBID** — An identifier of this object in the Configuration Database. Generated by Configuration Server and is unique within an object type. Identifiers of deleted objects are not used again. Read-only.
- **tenantDBID** — A unique identifier of the [CfgTenant](#) to which this table access is allocated. Mandatory. Once specified, cannot be changed.
- **name** — A pointer to the mnemonic name of table access. Mandatory. Must be unique within the Tenant.
- **type** — A type of table. Refer to [CfgTableType](#) of User Defined Variable Types. Mandatory. Once specified, cannot be changed.
- **description** — A list pointer to the description of table.
- **dbAccessDBID** — A unique identifier of the [CfgApplication](#) of type CFGDBServer (DB Access Point) through which the table can be accessed. Mandatory.

### Tip

There is no validation of the same tenantDBID reference in DatabaseAccessPoint when the DatabaseAccessPoint is assigned for [CfgTableAccess](#).

- **formatDBID** — A unique identifier of [CfgFormat](#) of this table. Once specified cannot be changed. The property is mandatory for all table types except CFGTTLogTable.  
See [CfgTableType](#).
- **dbTableName** — A pointer to the name of table in data base. Due to restrictions of some database engines the recommended length for property name is 1 to 12 characters. Mandatory.
- **isCachable** — An indicator of whether the table data shall be mirrored in application memory. See type [CfgFlag](#).
- **updateTimeout** — A timeout between updates of table data in application memory. Active if parameter **isCachable** set to true only.
- **state** — Current object state. Mandatory. Refer to [CfgObjectState](#).
- **userProperties** — A pointer to the list of user-defined properties. Parameter **userProperties** has the

following structure: Each key-value pair of the primary list (TKVList \*userProperties) uses the key for the name of a user-defined section, and the value for a secondary list, that also has the TKVList structure and specifies the properties defined within that section.

## Comments

One TableAccess can be dedicated to several objects of [CfgCallingList](#) type.

A TableAccess cannot be deleted as long as it is associated with at least one CallingList (see [CfgCallingList](#)).

## XML Representation

### Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgTableAccess>
  <DBID value="101" />
  <tenantDBID value="101" />
  <name value="Test_Table_Access" />
  <type value="1" />
  <description value="test list" />
  <dbAccessDBID value="119" />
  <formatDBID value="104" />
  <dbTableName value="Contact" />
  <isCachable value="1" />
  <updateTimeout value="0" />
  <state value="1" />
</CfgTableAccess>
```

## See Also

[CfgDeltaTableAccess](#)

[CfgFormat](#)