



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

CfgTenant

---

## Contents

- 1 CfgTenant
  - 1.1 Description
  - 1.2 Filter Keys
  - 1.3 Attributes
  - 1.4 Comments
  - 1.5 XML Representation
  - 1.6 See Also

# CfgTenant

## Description

A *Tenant* is a business whose customer interactions are enabled or enhanced through services offered by a third party, typically a telecommunications service provider.

From a functional standpoint, each tenant in a multi-tenant environment is a contact center (single or multi-site) completely equipped to process customer interactions.

From an architectural standpoint, however, most of the hardware and software that tenants use to enable or enhance those interactions belongs to the service provider.

## Filter Keys

Filter Name	Type	Description
dbid	int	A unique identifier of a tenant. If specified, Configuration Server will return information only about this tenant.
name	string	Name of a tenant. Shall be specified as a character string. If specified, Configuration Server will return information only about the tenant(s) with that name.
state	int	Current state of a tenant (see <a href="#">CfgObjectState</a> ). If specified, Configuration Server will return information only about tenants that are currently in this state.
all_tenants	int	If specified, Configuration Server will return information about all tenants including tenant with DBID=1.

## 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.
- **isServiceProvider** — An indicator of whether the tenant belongs to the Service Provider (see [CfgFlag](#)). Read-only (set automatically when a tenant is created).

- name — A pointer to name of the tenant. Mandatory. Must be unique within the Configuration Database.
- password — A pointer to the tenant password. Max length 64 symbols.
- address — Not in use.
- chargeableNumber — A pointer to the string value that is used for service charges of this tenant. Max length 64 symbols.
- tenantPersonDBID — Not in use.
- providerPersonDBID — Not in use.
- serviceInfo — Not in use.
- isSuperTenant — Not in use.
- tenantDBIDs — Not in use.
- parentTenantDBID — Configuration Server (with a schema version greater than 372) supports hierarchial multi-tenancy.
- 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.
- defaultCapacityRuleDBID — A unique identifier of the capacity rule ([CfgScript](#)) associated with this tenant.
- defaultContractDBID — A unique identifier of the cost contract ([CfgObjectiveTable](#)) associated with this tenant.

## Comments

Deletion of Tenant X will cause the following events set out in the order of arrival:

- Modification of tenantDBIDs of all applications associated with Tenant X (see [CfgApplication](#))
- Deletion of all agent groups of Tenant X
- Deletion of all place groups of Tenant X
- Deletion of all DN groups of Tenant X
- Deletion of all access groups of Tenant X
- Deletion of all scripts of Tenant X
- Deletion of all persons of Tenant X (see comments to [CfgPerson](#) for details)
- Deletion of all places of Tenant X
- Deletion of all switches of Tenant X (see comments to [CfgSwitch](#) for details)
- Deletion of all skills of Tenant X
- Deletion of all action codes of Tenant X
- Deletion of all stat tables of Tenant X

- Deletion of all stat days of Tenant X
- Deletion of all transactions of Tenant X
- Deletion of all fields of Tenant X
- Deletion of all formats of Tenant X
- Deletion of all filters of Tenant X
- Deletion of all table accesses of Tenant X
- Deletion of all treatments of Tenant X
- Deletion of all calling lists of Tenant X
- Deletion of all campaigns of Tenant X
- Deletion of all time zones of Tenant X
- Deletion of all voice prompts of Tenant X
- Deletion of all enumerators of Tenant X (see comments to [CfgEnumerator](#) for details)
- Deletion of all IVRs of Tenant X (see comments to [CfgIVR](#) for details)
- Deletion of all objective tables of Tenant X
- Deletion of all folders that had Tenant X defined as the parent object
- Deletion of Tenant X

The following tenant is pre-defined (scripted) in the Configuration Database before Configuration Server is started for the first time:

```
dbid = 1
isServiceProvider = CFGTrue
name             = "Environment"
password         = empty_string
isSuperTenant    = CFGFalse
state            = CFGEabled
```

This tenant (with DBID=1) cannot be deleted or modified in any way

The association between a solution and a tenant should be made using `assignedTenantDBID` within the [CfgService](#) object.

The tenant can not be deleted as long as contains persons serving as accounts for some server objects

## XML Representation

### Tip

This XML was created using the Configuration Server 7.5 schema.

## CfgTenant

---

```
<CfgTenant>
  <DBID value="101" />
  <isServiceProvider value="0" />
  <name value="Default" />
  <tenantPersonDBID value="0" />
  <providerPersonDBID value="0" />
  <isSuperTenant value="0" />
  <state value="1" />
  <defaultCapacityRuleDBID value="0" />
  <defaultContractDBID value="0" />
</CfgTenant>
```

## See Also

[CfgDeltaTenant](#)