



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

Platform SDK 9.0.x

12/29/2021

Table of Contents

| | |
|--|-----------|
| Introduction to the Configuration Layer Objects | 7 |
| List of Configuration Layer Objects | 10 |
| CfgAccessGroup | 11 |
| CfgActionCode | 14 |
| CfgAgentGroup | 17 |
| CfgAgentInfo | 19 |
| CfgAgentLogin | 20 |
| CfgAgentLoginInfo | 23 |
| CfgAlarmCondition | 24 |
| CfgAlarmEvent | 27 |
| CfgAppPrototype | 28 |
| CfgAppRank | 31 |
| CfgAppServicePermission | 32 |
| CfgApplication | 33 |
| CfgCallingList | 41 |
| CfgCallingListInfo | 44 |
| CfgCampaign | 45 |
| CfgCampaignGroup | 48 |
| CfgConnInfo | 51 |
| CfgDN | 53 |
| CfgDNAccessNumber | 58 |
| CfgDNGroup | 59 |
| CfgDNInfo | 62 |
| CfgDeltaAccessGroup | 63 |
| CfgDeltaActionCode | 64 |
| CfgDeltaAgentGroup | 65 |
| CfgDeltaAgentInfo | 66 |
| CfgDeltaAgentLogin | 67 |
| CfgDeltaAlarmCondition | 68 |
| CfgDeltaAppPrototype | 69 |
| CfgDeltaApplication | 70 |
| CfgDeltaCallingList | 72 |
| CfgDeltaCampaign | 73 |
| CfgDeltaCampaignGroup | 74 |
| CfgDeltaDN | 75 |

| | |
|-------------------------|-----|
| CfgDeltaDNGroup | 76 |
| CfgDeltaEnumerator | 77 |
| CfgDeltaEnumeratorValue | 78 |
| CfgDeltaField | 79 |
| CfgDeltaFilter | 80 |
| CfgDeltaFolder | 81 |
| CfgDeltaFormat | 82 |
| CfgDeltaGVPCustomer | 83 |
| CfgDeltaGVPIVRProfile | 84 |
| CfgDeltaGVPreSeller | 85 |
| CfgDeltaGroup | 86 |
| CfgDeltaHost | 87 |
| CfgDeltaIVR | 88 |
| CfgDeltaIVRPort | 89 |
| CfgDeltaObjectiveTable | 90 |
| CfgDeltaPerson | 91 |
| CfgDeltaPhysicalSwitch | 92 |
| CfgDeltaPlace | 93 |
| CfgDeltaPlaceGroup | 94 |
| CfgDeltaRole | 95 |
| CfgDeltaScheduledTask | 96 |
| CfgDeltaScript | 97 |
| CfgDeltaService | 98 |
| CfgDeltaSkill | 99 |
| CfgDeltaStatDay | 100 |
| CfgDeltaStatTable | 101 |
| CfgDeltaSwitch | 102 |
| CfgDeltaTableAccess | 103 |
| CfgDeltaTenant | 104 |
| CfgDeltaTimeZone | 105 |
| CfgDeltaTransaction | 106 |
| CfgDeltaTreatment | 107 |
| CfgDeltaVoicePrompt | 108 |
| CfgEnumerator | 109 |
| CfgEnumeratorValue | 113 |
| CfgField | 120 |
| CfgFilter | 125 |

| | |
|--------------------------------|-----|
| CfgFolder | 128 |
| CfgFormat | 132 |
| CfgGVPCustomer | 136 |
| CfgGVPIVRProfile | 138 |
| CfgGVPReseller | 140 |
| CfgGroup | 142 |
| CfgHost | 144 |
| CfgID | 147 |
| CfgIVR | 148 |
| CfgIVRPort | 150 |
| CfgOS | 152 |
| CfgObjectResource | 153 |
| CfgObjectiveTable | 154 |
| CfgObjectiveTableRecord | 157 |
| CfgPerson | 159 |
| CfgPhones | 164 |
| CfgPhysicalSwitch | 165 |
| CfgPlace | 167 |
| CfgPlaceGroup | 170 |
| CfgPortInfo | 173 |
| CfgRole | 174 |
| CfgRoleMember | 175 |
| CfgScheduledTask | 176 |
| CfgScript | 178 |
| CfgServer | 182 |
| CfgService | 183 |
| CfgServiceInfo | 187 |
| CfgSkill | 188 |
| CfgSkillLevel | 190 |
| CfgSolutionComponent | 191 |
| CfgSolutionComponentDefinition | 192 |
| CfgStatDay | 193 |
| CfgStatInterval | 197 |
| CfgStatTable | 198 |
| CfgSubcode | 201 |
| CfgSwitch | 202 |
| CfgSwitchAccessCode | 205 |

| | |
|---|------------|
| CfgTableAccess | 207 |
| CfgTenant | 210 |
| CfgTimeZone | 214 |
| CfgTransaction | 221 |
| CfgTreatment | 223 |
| CfgVoicePrompt | 229 |
| List of Configuration Layer Enumerations | 232 |
| CfgAccessGroupType | 233 |
| CfgActionCodeType | 234 |
| CfgAlarmCategory | 235 |
| CfgAppComponentType | 236 |
| CfgAppType | 237 |
| CfgCallActionCode | 244 |
| CfgChargeType | 245 |
| CfgCTILinkType | 246 |
| CfgDIDGroupType | 247 |
| CfgDNGroupType | 248 |
| CfgDNRegisterFlag | 250 |
| CfgDNType | 251 |
| CfgDataType | 255 |
| CfgDialMode | 259 |
| CfgEnumeratorObjectType | 260 |
| CfgEnumeratorType | 261 |
| CfgErrorType | 262 |
| CfgEventType | 268 |
| CfgFieldType | 275 |
| CfgFilterType | 277 |
| CfgFlag | 281 |
| CfgFolderClass | 282 |
| CfgGroupType | 283 |
| CfgHAType | 284 |
| CfgHostType | 285 |
| CfgIVRProfileType | 286 |
| CfgIVRType | 287 |
| CfgLanguage | 289 |
| CfgLinkType | 290 |
| CfgMediaType | 294 |

| | |
|-----------------------|-----|
| CfgOSType | 295 |
| CfgObjectState | 296 |
| CfgObjectType | 297 |
| CfgObjectiveTableType | 304 |
| CfgOperationMode | 305 |
| CfgOperationalMode | 306 |
| CfgOptimizationMethod | 307 |
| CfgPermissions | 308 |
| CfgPersonType | 309 |
| CfgRank | 310 |
| CfgRecActionCode | 311 |
| CfgResourceType | 313 |
| CfgRouteType | 315 |
| CfgScriptType | 317 |
| CfgSelectionMode | 319 |
| CfgSolutionType | 320 |
| CfgStartupType | 322 |
| CfgStatDayType | 324 |
| CfgStatTableType | 325 |
| CfgSwitchType | 326 |
| CfgTableType | 330 |
| CfgTargetType | 331 |
| CfgTaskType | 332 |
| CfgTraceMode | 333 |
| CfgTransactionType | 334 |
| GctiCallState | 335 |
| GctiContactType | 337 |
| GctiRecordStatus | 338 |
| GctiRecordType | 339 |

Introduction to the Configuration Layer Objects

Important

Content on this page comes from the Platform SDK [Developer's Guide](#), but is copied here to provide a better understanding of the reference material included in this guide.

The Genesys Configuration Layer is a database containing information about the objects in your contact center environment. You may need to get information about these objects. You may also want to add, update, or delete them. The Configuration Platform SDK gives you the means to do that.

This article contains information that is common to all of these Configuration Layer objects.

General Parameters

The following parameters are common to objects of all types. They will not be described again in the listings for individual objects.

- **DBID** — An identifier of this object in the Configuration Database. Generated by Configuration Server, it is unique within an object type. Identifiers of deleted objects are not used again. Read-only.
- **state** — Current object state. Mandatory. Refer to `CfgObjectState` in section [Variable Types](#).

Tip

Change in the state of a parent object will cause the states of all its child objects to change accordingly. Configuration Server will provide a notification for each elementary change. Changing the state of a parent object will not be allowed unless the client application has privileges to change all of the child objects of this parent object.

- **userProperties** — In objects, a pointer to the list of user-defined properties. In delta objects, a pointer to a list of user-defined properties added to the existing list. 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. Each key-value pair of the secondary list uses the key for the name of a user-defined property, and the value for its current setting. User properties can be defined as variables of integer, character, or binary type. Names of sections must be unique within the primary list. Names of properties must be unique within the secondary list.

Tip

Configuration Server is not concerned with logical meanings of user-defined sections, properties, or their values.

- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties` above. A user-defined property is deleted by specifying the name of the section that this property belongs to, and the name of the property itself with any value. A whole section is deleted by specifying the name of that section and an empty secondary list.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties` above. A value of a user-defined property is changed by specifying the name of the section that this property belongs to, the name of the property itself, and a new value of that property.
- `flexibleProperties` — In objects, a pointer to the list of additional properties. In delta objects, a pointer to a list of user-defined properties added to the existing list. This parameter has the following structure: Each key-value pair of the primary list (`TKVList * flexibleProperties`) uses the key for the name of the section, and the value for a secondary list, that also has the `TKVList` structure and specifies either properties defined within that section or another section name. Each key-value pair of the secondary list uses the key for the name of a property, and the value for its current setting. Properties can be defined as variables of integer, character, or binary type or as the name of another list of properties. Names of sections must be unique within the primary list. Names of properties must be unique within the list. The data structure within the `flexibleProperties` property is object-type specific and hard-coded within Configuration Server. Each key-value in the `TKVList * flexibleProperties` is controlled and processed by Configuration Server only in the same manner as any other property in contrast with user-properties the contents of which are not Configuration Server concerned. If the structure of the property's `Extension` is not specified, the value is `NULL`. For more information, see the detailed object descriptions in this document.

Configuration Object Association

Configuration Objects can be associated with each other in a number of different ways that can be generally classified as follows:

- Parent-child relationship, where a child object cannot be created without a parent and will be deleted automatically if its parent object is deleted. Most of the object types will have an explicit reference to their parents which is marked with an asterisk in the specification below. For the object types that do not have such a reference, it is implied that their parent is the Service Provider (that is, the imaginary tenant with `DBID = 1`).
- Exclusive association, where an object cannot be associated in the same manner with more than one other object.
- Non-exclusive association, where an object can be associated in the same manner with more than one other object. Unless expressly noted otherwise, a reference to the `DBID` of another object without an asterisk indicates a non-exclusive assignment.

The parameters of all object-related structures are optional unless otherwise noted. However, all variables of character type must be initialized at the time an object is created. The variables of character type that are not mandatory may be initialized with an empty string (the recommended

default value unless otherwise noted). The variables of character type that are mandatory may not be initialized with an empty string. Variables of character type may accept values of up to 255 symbols in length unless otherwise noted. The recommended default value for optional parameters of other types is zero or NULL, unless otherwise noted.

Filters

Filters are used to specify more precisely the kind of information that the client application is interested in. Filters reduce both volumes of data communicated by Configuration Server and data-processing efforts on the client side. Filters are structured as key-value pairs where the value of each key defines a certain condition of data selection. Filter keys are defined as variables of integer type unless otherwise noted.

Important

Although your application can use "and" to combine multiple filters when retrieving a set of matching configuration objects, specifying a DBID value as one of the filters causes all other filters in that request to be ignored. This is by design, as only a single configuration object can match the specified DBID value. However, this behavior could create unexpected results if your application intended to use filters as a method for checking whether a known configuration object also matches additional filter values.

Here is a list of common filter types:

- `folder_dbid` — A unique identifier of a folder. If specified, Configuration Server will return information only about objects of specific type located under specified folder. See also the description of the `ConfGetObjectInfo` function.
- `delegate_dbid` — A unique identifier of an account on behalf of which current query is to be executed. Produced result set will be calculated using a superposition of the registered account permissions and that passed in `delegate_dbid` filter. Must be used in conjunction with `delegate_type` filter in order to specify account type (`CFGPerson` or `CFGAccessGroup`).
- `delegate_type` — Object type of the account (`CFGPerson` or `CFGAccessGroup`) on behalf of which the current query is to be executed. Must be used in conjunction with `delegate_dbid`.
- `object_path` — A flag that causes Configuration Server to return a full path of the object in the folder hierarchy for every object in the result set. The path string will be returned in the `cfgDescription` field of the `CFGObjectInfo` event.
- `cmp_insensitive` — A flag that causes Configuration Server to perform case-insensitive comparison of string values in the filter. Supported from Configuration Server 7.2.000.00.
- `read_folder_dbid` — A flag that causes Configuration Server to return a Folder DBID for every object in the result set. The folder will be returned in the `cfgExtraInfo3` field of the `CFGObjectInfo` event. Supported from Configuration Server 7.2.000.00.

List of Configuration Layer Objects

The following table provides a convenient list of Configuration Layer Objects that are available. For more information, refer to [Introduction to the Configuration Layer Objects](#).

| | | | |
|-------------------------|-------------------------|-------------------------|--------------------------------|
| CfgAccessGroup | CfgDeltaCampaign | CfgDeltaTableAccess | |
| CfgActionCode | CfgDeltaCampaignGroup | CfgDeltaTenant | |
| CfgAgentGroup | CfgDeltaDN | CfgDeltaTimeZone | |
| CfgAgentInfo | CfgDeltaDNGroup | CfgDeltaTransaction | CfgRole |
| CfgAgentLogin | CfgDeltaEnumerator | CfgDeltaTreatment | CfgRoleMember |
| CfgAgentLoginInfo | CfgDeltaEnumeratorValue | CfgDeltaVoicePrompt | CfgScheduledTask |
| CfgAlarmCondition | CfgDeltaField | CfgEnumerator | CfgScript |
| CfgAlarmEvent | CfgDeltaFilter | CfgEnumeratorValue | CfgServer |
| CfgAppPrototype | CfgDeltaFolder | CfgField | CfgService |
| CfgAppRank | CfgDeltaFormat | CfgFilter | CfgServiceInfo |
| CfgAppServicePermission | CfgDeltaGVPCustomer | CfgFolder | CfgSkill |
| CfgApplication | CfgDeltaGVPIVRProfile | CfgFormat | CfgSkillLevel |
| CfgCallingList | CfgDeltaGVPReseller | CfgGVPCustomer | CfgSolutionComponent |
| CfgCallingListInfo | CfgDeltaGroup | CfgGVPIVRProfile | CfgSolutionComponentDefinition |
| CfgCampaign | CfgDeltaHost | CfgGVPReseller | CfgStatDay |
| CfgCampaignGroup | CfgDeltaIVR | CfgGroup | CfgStatInterval |
| CfgConnInfo | CfgDeltaIVRPort | CfgHost | CfgStatTable |
| CfgDN | CfgDeltaObjectiveTable | CfgID | CfgSubcode |
| CfgDNAccessNumber | CfgDeltaPerson | CfgIVR | CfgSwitch |
| CfgDNGroup | CfgDeltaPhysicalSwitch | CfgIVRPort | CfgSwitchAccessCode |
| CfgDNInfo | CfgDeltaPlace | CfgOS | CfgTableAccess |
| CfgDeltaAccessGroup | CfgDeltaPlaceGroup | CfgObjectResource | CfgTenant |
| CfgDeltaActionCode | CfgDeltaRole | CfgObjectiveTable | CfgTimeZone |
| CfgDeltaAgentGroup | CfgDeltaScheduledTask | CfgObjectiveTableRecord | CfgTransaction |
| CfgDeltaAgentInfo | CfgDeltaScript | CfgPerson | CfgTreatment |
| CfgDeltaAgentLogin | CfgDeltaService | CfgPhones | CfgVoicePrompt |
| CfgDeltaAlarmCondition | CfgDeltaSkill | CfgPhysicalSwitch | |
| CfgDeltaAppPrototype | CfgDeltaStatDay | CfgPlace | |
| CfgDeltaApplication | CfgDeltaStatTable | CfgPlaceGroup | |
| CfgDeltaCallingList | CfgDeltaSwitch | CfgPortInfo | |

CfgAccessGroup

Description

Access Groups are groups of Persons who need to have the same set of permissions with respect to Configuration Database objects

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|---|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the access groups that belong to this tenant. |
| person_dbid | int | A unique identifier of a Person. If specified, Configuration Server will return information only about the access groups this Person is assigned to. |
| state | int | Current state of an access group (see CfgObjectState). If specified, Configuration Server will return information only about access groups that are currently in this state. |
| name | string | Name of an access group. Shall be specified as a character string. If specified, Configuration Server will return information only about the access group(s) with that name. |
| dbid | int | A unique identifier of a group. If specified, Configuration Server will return information only about this group. |

Attributes

- `groupInfo` — A pointer to the structure [CfgGroup](#) containing general information about this group. Mandatory.

- **memberIDs** — A pointer to the list of the Persons that form this group (every item of this list is structured as **CfgID**).
- **type** — Type of this Access Group. Once specified, cannot be changed. Refer to **CfgAccessGroupType**.

Comments

The name of a access group must be unique within the tenant, but can coincide with the names of dn groups, place groups and agent groups of the same tenant.

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgAccessGroup>
  <CfgGroup>
    <DBID value="102" />
    <tenantDBID value="1" />
    <name value="Group" />
    <capacityTableDBID value="0" />
    <quotaTableDBID value="0" />
    <state value="1" />
    <capacityRuleDBID value="0" />
    <siteDBID value="0" />
    <contractDBID value="0" />
  </CfgGroup>
  <memberIDs>
    <CfgID>
      <CSID value="0" />
      <DBID value="119" />
      <type value="3" />
    </CfgID>
    <CfgID>
      <CSID value="0" />
      <DBID value="120" />
      <type value="3" />
    </CfgID>
  </memberIDs>
  <type value="3" />
</CfgAccessGroup>
```

See Also

- [CfgDeltaAccessGroup](#)

-
- [CfgPerson](#)

CfgActionCode

Description

Action coding enables agents to report the business results of customer interactions, as well as to explain the reasons for certain operations.

After you select an appropriate code from a menu of predefined Action Codes, the code is passed along with its related request. The code then returns with the event, which indicates that the request has been successfully processed. Eventually, the code is stored in the reporting database.

You can supplement each Action Code with a number of Subcodes that more precisely characterize the reasons for a certain action.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|---|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the action codes that belong to this tenant. |
| code_type | int | Type of the action code (see CfgActionCodeType). If specified, Configuration Server will return information only about the action codes of this type. |
| state | int | Current state of an action code (see CfgObjectState). If specified, Configuration Server will return information only about action codes that are currently in this state. |
| dbid | int | A unique identifier of an action code. If specified, Configuration Server will return information only about this action code. |
| name | string | Name of an action code. Shall be specified as a character string. If specified, Configuration Server will return information only about the action code(s) with that name. |

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 **Tenant** that this action code belongs to. Mandatory. Once specified, cannot be changed.
- **name** — A pointer to the name of the action code. Mandatory. Must be unique within the action code type specified below for the given tenant.
- **type** — Type of the action code. See type `CfgActionCodeType`. Mandatory. Once specified, cannot be changed.
- **code** — A pointer to the index or abbreviation of the action code. Mandatory. Must be unique within the action code type specified above for the given tenant.
- **subcodes** — A pointer to the list of the subcodes defined within the action code (every item of this list is structured as `CfgSubcode`). When used as an entry in `CfgDeltaActionCode` (see below), it is a pointer to a list of subcodes added to the existing list.
- **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.

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgActionCode>
  <DBID value="101" />
  <tenantDBID value="1" />
  <name value="Action" />
  <type value="9" />

  <state value="1" />
  <userProperties>
    <list_pair key="Property">
      <str_pair key="key" value="value" />
    </list_pair>
  </userProperties>
</CfgActionCode>
```

See Also

[CfgDeltaActionCode](#)

CfgAgentGroup

Description

Groups of Agents are typically set up to provide particular sets of contact center services.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|---|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the agent groups that belong to this tenant. |
| person_dbid | int | A unique identifier of a person. If specified, Configuration Server will return information only about the agent groups this person is assigned to. |
| state | int | Current state of an agent group (see CfgObjectState). If specified, Configuration Server will return information only about agent groups that are currently in this state. |
| name | string | Name of an agent group. Shall be specified as a character string. If specified, Configuration Server will return information only about the agent group(s) with that name. |
| dbid | int | A unique identifier of a group. If specified, Configuration Server will return information only about this group. |

Attributes

- `groupInfo` — A pointer to the structure [CfgGroup](#) containing general information about this group. Mandatory.
- `agentDBIDs` — A pointer to the list of identifiers of the [Agents](#) that form this group.

Comments

The name of an agent group must be unique within the tenant, but can coincide with the name of either a place group or a DN group of the same tenant.

The name of an agent group cannot be changed until there is at least one agent listed in this group. See `agentDBIDs` property

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

- Modification of `campaignGroups` of all campaigns that included Agent Group X
- Deletion of Agent Group X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgAgentGroup>
  <CfgGroup>
    <DBID value="107" />
    <tenantDBID value="101" />
    <name value="AGroup" />
    <capacityTableDBID value="0" />
    <quotaTableDBID value="0" />
    <state value="1" />
    <capacityRuleDBID value="0" />
    <siteDBID value="0" />
    <contractDBID value="0" />
  </CfgGroup>
  <agentDBIDs>
    <DBID value="120" />
    <DBID value="121" />
  </agentDBIDs>
</CfgAgentGroup>
```

See Also

- [CfgDeltaAgentGroup](#)
- [CfgPerson](#)

CfgAgentInfo

Description

CfgAgentInfo contains information about an Agent.

Attributes

- `placeDBID` — A unique identifier of the [Place](#) assigned to this agent by default. The place must belong to the same tenant as the person in question unless this Agent belongs to the tenant Environment (with `DBID=1`)
- `skillLevels` — A pointer to the list of the agent's skill levels (every item of this list is structured as [CfgSkillLevel](#)).
When used as an entry in [CfgDeltaAgentInfo](#) (see below), it is a pointer to a list of skill levels added to the existing list.
- `agentLogins` — A pointer to the list of the agent logins assigned to this agent (every item of this list is structured as [CfgAgentLoginInfo](#)).
When used as an entry in [CfgDeltaAgentInfo](#) (see below), it is a pointer to a list of agent logins added to the existing list.
- `capacityRuleDBID` — A unique identifier of the capacity rule ([CfgScript](#)) associated with this agent.
- `siteDBID` — A unique identifier of Site ([CfgFolder](#)) with which this Agent is associated.
- `contractDBID` — A unique identifier of Cost Contract ([CfgObjectiveTable](#)) with which this Agent is associated.

CfgAgentLogin

Description

Agent Logins are unique codes defined within a Switch and assigned to agents. They identify which Agent is working at which Place during a particular working session.

Configuration of Agent Logins in the Configuration Database must exactly match the configuration of those Agent Logins in the switching system. Before adding or deleting a particular Agent Login, make sure that the same change was made in the database of the switching system.

When you specify Agent Logins as objects in a Switch, they are not associated with any particular agents.

Filter Keys

| Filter Name | Type | Description |
|----------------|--------|---|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the agent logins that belong to this tenant. |
| switch_dbid | int | A unique identifier of a switch. If specified, Configuration Server will return information only about the agent logins that belong to this switch. |
| state | int | Current state of an agent login (see CfgObjectState). If specified, Configuration Server will return information only about agent logins that are currently in this state. |
| login_code | string | Agent login code. Shall be specified as a character string. If specified, Configuration Server will return information only about the agent login(s) with this code. |
| dbid | int | A unique identifier of an agent login. If specified, Configuration Server will return information only about this agent login. |
| no_person_dbid | int | Configuration Server will return information only about agent |

| Filter Name | Type | Description |
|-------------|------|---|
| | | logins that currently are not associated with any person. |

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.
- **switchDBID** — A unique identifier of the Switch to which this agent login belongs. Mandatory. Once specified, cannot be changed.
- **tenantDBID** — A unique identifier of the Tenant to which this agent login belongs. Read-only (set automatically according to the current value of tenantDBID of the switch specified in switchDBID). See type CfgSwitch.
- **loginCode** — A pointer to the agent login code. Mandatory. Must be unique within the switch. Once specified, cannot be changed.
- **state** — Current object state. Mandatory. Refer to CfgObjectState
- **override** — The number used as a substitute of a regular agent login in certain types of routing.
- **useOverride** — An indicator of whether the override value shall be used instead of the loginCode value for accessing this agent login in certain types of routing. Recommended to be set to CFGTrue by default. See type CfgFlag.
- **switchSpecificType** — An integer that corresponds to a combination of switch-specific settings for this agent login. Cannot be set to a zero or negative value.
- **password** — A pointer to the agent login password.

Comments

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

- Modification of agentLogins of the agent who had Agent Login X assigned
- Deletion of Agent Login X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgAgentLogin>
  <DBID value="101" />
  <switchDBID value="102" />
  <tenantDBID value="1" />
  <loginCode value="001" />
  <state value="1" />
  <userProperties>
    <list_pair key="Property">
      <str_pair key="key" value="value" />
    </list_pair>
  </userProperties>
  <useOverride value="2" />
  <switchSpecificType value="1" />
</CfgAgentLogin>
```

See Also

[CfgDeltaAgentLogin](#)

CfgAgentLoginInfo

Description

Agent login information.

Attributes

- agentLoginDBID — A unique identifier of the **Agent Login** identifier. Mandatory. Once specified, cannot be changed. The switch that this login identifier belongs to must be assigned to the tenant that this agent belongs to. The same value cannot be repeated within one list. One agent login cannot be assigned to more than one agent.
- wrapupTime — Wrap-up time in seconds associated with this login identifier. Cannot be a negative value.

CfgAlarmCondition

Description

Alarm Conditions specify events that you might want to know about and manage as soon as they occur.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|---|
| state | int | Current state of an alarm condition (see CfgObjectState). If specified, Configuration Server will return information only about alarm conditions that are currently in this state. |
| name | string | Name of an alarm condition. Shall be specified as a character string. If specified, Configuration Server will return information only about the alarm condition(s) with that name. |
| dbid | int | A unique identifier of an alarm condition. If specified, Configuration Server will return information only about this alarm condition. |

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.
- **name** — A pointer to the name of the alarm condition. Mandatory. Must be unique within the Configuration Database.
- **description** — A pointer to the description of the alarm condition.
- **category** — The category of the alarm condition. Mandatory. See type [CfgAlarmCategory](#). Default value is `CFGACMajor`.
- **alarmDetectEvent** — A pointer to the [CfgAlarmEvent](#) structure which is used to describe a log event upon which an alarm based on this alarm condition should be detected. Mandatory.

- `alarmRemovalEvent` — A pointer to the `CfgAlarmEvent` structure which is used to describe a log event upon which an alarm based on this alarm condition should be removed.
- `alarmDetectScriptDBID` — A unique identifier of a script which describes the logic to be applied to detect an alarm based on this alarm condition. Only a script whose type is `CFGAlarmDetection` can be specified. Reserved for future use. See `CfgScript`.
- `clearanceTimeout` — The period of time, in seconds, upon which an alarm based on this alarm condition has to be cleared since the moment it was detected. Default value is 24 hours ($24*60*60 = 86400$).
- `reactionScriptDBIDs` — A pointer to a list of identifiers of the `CfgScript` that describe reactions to an alarm based on this alarm condition. Only scripts whose type is `CFGAlarmReaction` can be specified. When used as an entry in `CfgDeltaAlarmCondition` (see below), it is a pointer to a list of identifiers of the scripts added to the existing list. See `CfgScript`.
- `isMasked` — Determines whether an alarm which is based on this alarm condition should be communicated to Solution Control Interface and reactions to the alarm should be performed. This corresponds to the default value of `CFGFalse`. Mandatory. See `CfgFlag`.
- `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.
- `clearanceScriptDBIDs` — A pointer to a list of identifiers of the `CfgScript` that describe clearance to an alarm based on this alarm condition. Only scripts whose type is `CFGAlarmReaction` can be specified. When used as an entry in `CfgDeltaAlarmCondition` (see below), it is a pointer to a list of identifiers of the scripts added to the existing list. See `CfgScript`.

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgAlarmCondition>
  <DBID value="101" />
  <name value="An Alarm" />
  <description value="Description." />
  <category value="2" />
  <CfgDetectEvent>
    <logEventID value="2001" />
    <selectionMode value="3" />
    <appType value="0" />
    <appDBID value="0" />
  </CfgDetectEvent>
  <CfgRemovalEvent>
    <logEventID value="2002" />
    <selectionMode value="3" />
    <appType value="0" />
    <appDBID value="0" />
  </CfgRemovalEvent>
  <alarmDetectScriptDBID value="0" />

```

```
<clearanceTimeout value="172800" />  
<isMasked value="1" />  
<state value="1" />  
</CfgAlarmCondition>
```

See Also

[CfgDeltaAlarmCondition](#)

[CfgScript](#)

[CfgApplication](#)

CfgAlarmEvent

Description

An Alarm Event specifies the event that triggers an alarm.

Attributes

- `logEventID` — An identifier of a specific log event of interest. See Log API for more information.
- `selectionMode` — Determines the event selection mode that is used for analysis. Default value is `CFGSMByAny`. See [CfgSelectionMode](#).
- `appType` — An application type to describe the type of applications which can generate the log event with the specified `logEventID`. Should only be used if the value of `useAppType` is set to `CFGTrue`. See [CfgAppType](#).
- `appDBID` — A unique identifier of an [CfgApplication](#) which generates the log event with the specified `logEventID`. Could only be used if the value of `useAppType` is set to `CFGFalse`. Optional.

CfgAppPrototype

Description

An application prototype.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|---|
| app_type | int | Type of the application (see type CfgAppType). If specified, Configuration Server will return information only about the application prototypes that relates to the applications of this type. |
| state | int | Current state of an application prototype (see CfgObjectState). If specified, Configuration Server will return information only about application prototypes that are currently in this state. |
| name | string | Name of an application prototype. Shall be specified as a character string. If specified, Configuration Server will return information only about the application prototype with that name. |
| dbid | int | A unique identifier of an application. If specified, Configuration Server will return information only about this application prototype. |
| version | string | A version of the application. Shall be specified as a character string. If specified, Configuration Server will return information only about application prototypes with that version. |

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.
- **name** — A pointer to the name of the application prototype. Mandatory. Once specified, cannot be changed. Must be unique within the Configuration Database.
- **type** — Type of the application. Mandatory. Once specified, cannot be changed. See [CfgAppType](#).
- **version** — A pointer to the application version. Once specified, cannot be changed.
- **options** — A pointer to the list of application-specific configuration options with default values where appropriate (see the comments below). When used as an entry in `CfgDeltaApplication`, it is a pointer to a list of options added to the existing list.
- **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

An application prototype cannot be deleted as long as there is at least one application associated with it.

Parameter `options` has the following structure: Each key-value pair of the primary list (`TKVList *options`) uses the key for the name of a configuration section, and the value for a secondary list, that also has the `TKVList` structure and specifies the configuration options defined within that section. Each key-value pair of the secondary list uses the key for the name of a configuration option, and the value for its current setting. Configuration options can be defined as variables of integer, character, or binary type. Names of sections must be unique within the primary list. Names of options must be unique within the secondary list.

Tip

Configuration Server is not concerned with logical meanings of application-specific configuration sections, options, or their values.

An application prototype of `CFGConfigServer` type with `DBID` set to 99 shall be pre-defined (scripted) in the Configuration Database before Configuration Server is started for the first time. The object that represents this application prototype cannot be deleted.

An application prototype of `CFGSCCE` type with `DBID` set to 100 shall be pre-defined (scripted) in the Configuration Database before Configuration Server is started for the first time. The object that represents this application prototype cannot be deleted.

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgAppPrototype>
  <DBID value="99" />
  <name value="Prototype" />
  <type value="21" />
  <version value="7.5" />
  <state value="1" />
</CfgAppPrototype>
```

See Also

[CfgDeltaAppPrototype](#)

CfgAppRank

Description

Ranks allow Applications to control which of their functions to make available to the currently logged-in Person.

Attributes

- `appType` — Type of the application this rank relates to. May make sense for certain applications of the GUI type. (See [CfgAppType](#) and the comment below.) Mandatory. Once specified, cannot be changed. The same value cannot be repeated within one list.
- `appRank` — Application rank. Rank `CFGTenantAdministrator` cannot be assigned to a person whose `tenantDBID` is 1. Ranks `CFGServiceAdministrator` and `CFGSuperAdministrator` cannot be assigned to a person whose `tenantDBID` is not 1. See type [CfgRank](#).

Comments

Due to the introduction of a flexible access control system in Configuration Server version 5.1.100, the only purpose left for application ranks is to control what functionality of a certain application is available to the currently logged-on person. The decision on whether to use rank-based access to application's functions, and what functions to block/enable for what rank shall be determined exclusively by the feature requirements to that application type and made part of the functional specification for that application type. Level of access of a particular person to the Configuration Database objects does not depend in any way on the set of application ranks of that person.

CfgAppServicePermission

Description

Attributes

- `appType` — Type of the application this permission mask relates to. See [CfgAppType](#). Mandatory. Once specified, cannot be changed. The same value cannot be repeated within one list.
- `permissionMask` — Application permission mask. Recommended to be set to “all ones” by default.

Comments

The only purpose of service permission masks is to control what functionality of a certain application is available within the service where it is used. The decision on whether to use service-based access to application's functions and availability of what functions to make service-dependent shall be determined exclusively by the feature requirements to that application type and made part of the functional specification for that application type. Level of access of a particular application to the Configuration Database objects does not depend in any way on settings of the above parameters.

CfgApplication

Description

Applications are the various Genesys software programs that serve a contact center. There are two types of Applications: GUI-based Applications and daemon Applications.

Filter Keys

| Filter Name | Type | Description |
|---------------------|------|---|
| app_type | int | Type of the application (see CfgAppType). If specified, Configuration Server will return information only about the applications of this type. |
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the applications that are associated with this tenant. |
| is_server | int | An indicator of whether this application can be a server to some other applications. Depending on value specified, Configuration Server will return information only about the application(s) that are either servers or non-servers. See CfgFlag . |
| server_dbid | int | A unique identifier of an application. If specified, Configuration Server will return information only about the applications that are clients to this application. |
| primary_server_dbid | int | A unique identifier of an application. If specified, Configuration Server will return information only about the application that is backup to this application. |
| backup_server_dbid | int | A unique identifier of an application. If specified, Configuration Server will return information only about the |

| Filter Name | Type | Description |
|--------------------|------|--|
| | | application that is primary to this application. |
| app_prototype_dbid | int | A unique identifier of an application prototype. If specified, Configuration Server will return information only about the applications that are based on this prototype. |
| account_type | int | Type of the object that may be used as an account for a daemon application (see CfgObjectType). Makes sense only if used with the filter account_dbid (see below). If both account_type and account_dbid are specified, Configuration Server will return information only about the applications associated with this account. Such information will only be provided to the clients that have privileges to read access control list of this application. |
| account_dbid | int | A unique identifier of the object that may be used as an account for a daemon application (see type CfgObjectType). Makes sense only if used with the filter account_type (see below). Makes sense only if used with the filter account_dbid (see below). If both account_type and account_dbid are specified, Configuration Server will return information only about the applications associated with this account. Such information will only be provided to the clients that have privileges to read access control list of this application. |
| host_dbid | int | A unique identifier of a host. If specified, Configuration Server will return information only about the applications currently assigned to this host. |
| port | int | A server communication port. If specified, Configuration Server will return information only about the applications currently registered at ports with this number. Consider using this filter with filter host_dbid (see |

| Filter Name | Type | Description |
|--------------------|--------|--|
| | | above). |
| state | int | Current state of an application (see CfgObjectState). If specified, Configuration Server will return information only about applications that are currently in this state. |
| name | string | Name of an application. Shall be specified as a character string. If specified, Configuration Server will return information only about the application with that name. |
| dbid | int | A unique identifier of an application. If specified, Configuration Server will return information only about this application. |
| same_host_and_port | int | Configuration Server will return information only about applications currently registered on same host and port. Filter is intended to avoid the configuration collisions. |
| switch_dbid | int | A unique identifier of the switch. If specified, Configuration Server will return information only about T-Servers/HAProxies associated with this switch (see flexibleProperties above). The filter makes sense for application types T-Server and High Availability Proxy (CFGTServer; CFGHAProxy.) |
| version | string | A version of the application. Shall be specified as a character string. If specified, Configuration Server will return information only about applications with that version. |
| no_switch_dbid | int | If specified, Configuration Server will return information only about T-Servers/ HAProxies that are not associated with any switches (see flexibleProperties above). The filter makes sense for application types T-Server and High Availability Proxy (CFGTServer; CFGHAProxy.) |
| no_client_dbid | int | If specified, Configuration Server will return information only about applications/servers which do not have any clients (there is no |

| Filter Name | Type | Description |
|--------------|------|---|
| | | connection to this applications) configured. |
| startup_type | int | Startup type of the application (see CfgStartupType). If specified, Configuration Server will return information only about the applications of this startup type. |

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.
- **name** — A pointer to the name of the application. Mandatory. Must be unique within the Configuration Database.
- **type** — Type of the application. Mandatory. Set automatically according to the value of type of the application prototype specified in `appPrototypeDBID` or explicitly during the creation time. Once specified, cannot be changed. See [CfgAppType](#).
- **password** — A pointer to the application password. Max length 64 symbols. Not used in 5.1.
- **version** — A pointer to the application version. Mandatory. Set automatically according to the value of version of the application prototype specified in `appPrototypeDBID` or explicitly.
- **appServerDBIDs** — A pointer to the list of structures of type [CfgConnInfo](#). [CfgConnInfo](#) includes:
 - the identifier of the server this application shall connect to as a client when it is started
 - this specific connection properties.

Applications can be added to the `appServerDBIDs` structure only if:

 - they are daemons, and
 - their `hostDBID` and port have been specified.

Tip

Regarding backward compatibility, for applications using Configuration Libraries release 5.1.xxx and 5.1.5xx, the `appServerDBIDs` will include the list of `ServerDBIDs` only.

- **tenantDBIDs** — A pointer to the list of identifiers of the [Tenants](#) that are served by this application. Makes sense only for applications of the daemon type. For applications of `CFGTServer` and `CFGHAProxy` type, can contain only one tenant. A tenant can be added to this list only if the account that the application is associated with has at least *read-only access* to this tenant. When used as an entry in `CfgDeltaApplication`, it is a pointer to a list of identifiers of the tenants added to the existing list.
- **isServer** — An indicator of whether this application can be a server to some other applications. Read-only (set automatically according to the value of type above). See [CfgFlag](#).

-
- `serverInfo` — A pointer to the structure containing server-specific information. Can be specified if, according to the value specified for the type above, the application is a daemon and must be set to NULL otherwise. Once specified, cannot be set to NULL. See [CfgServer](#).
 - `options` — A pointer to the list of application-specific configuration options (see the comments below). When used as an entry in `CfgDeltaApplication`, it is a pointer to a list of options added to the existing list.
 - `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.
 - `appPrototypeDBID` — A unique identifier of an application prototype this application is based on. Optional. See [CfgAppPrototype](#). The association with application prototype could be specified at moment of creation of application object only.
 - `flexibleProperties` — A pointer to the list of additional properties. See comments section at the end of this article. Only described below options can be added to this property. This field can not be changed as long as this server remains a backup for some other server (see [CfgServer](#) structure) and can only be non-empty for the applications of type `CFGTServer` and `CFGHAProxy`
 - `workDirectory` — Working directory for the application. Must be specified if, according to the value specified in `isServer` property, the application is a server and optional otherwise. See comments.
 - `commandLine` — The name of executable to be used to start the application. Must be specified if, according to the value specified in `isServer` property, the application is a server and optional otherwise. See comments.
 - `autoRestart` — Indicates whether the application should be automatically restarted by Local Control Agent after its crash. Mandatory. Recommended to be set to `CFGTrue` by default. See comments. See [CfgFlag](#).
 - `startupTimeout` — A period of time within which the application is expected to be completely started. Must be specified if, according to the value specified in `isServer` property, the application is a server and optional otherwise. Default value is 90 seconds. See comments.
 - `shutdownTimeout` — A period of time within which the application is expected to be completely shut down. Must be specified if, according to the value specified in `isServer` property, the application is a server and optional otherwise. Default value is 90 seconds. See comments.
 - `redundancyType` — Defines the HA type if this application is considered as server. Mandatory. Default is `CFGHTCoIdStandby`. See comments. See [CfgHAType](#).
 - `isPrimary` — A role of application within HA/redundancy group. Must be considered in association with `redundancyType` property. Default is `CFGTrue`. Read-only (set in accordance with the current role within the HA/redundancy group). See comments. See [CfgFlag](#).
 - `startupType` — A type of application startup. Indicates whether this application have to be started by Management Layer. See type [CfgStartupType](#). Read-only. The property is accessible via API only and not shown by Configuration Manager. Specified during application prototype definition automatically according following: The value is associated with application type `CfgAppType`. The value for the applications of `CFGDBServer` and `CFGApplicationCluster` type is set to `CFGSUTDisabled`, for other applications of server type is set to `CFGSUTAutomatic`. For the applications of non-server type is set to `CFGSUTDisabled`.
 - `commandLineArguments` — A pointer to the additional arguments to be used to start the application. Optional.
 - `portInfos` — A pointer to the list of structures of type [CfgPortInfo](#) containing information about listening ports for this Server application. When used as an entry in [CfgDeltaApplication](#), it is a pointer to a list of
-

port infos added to the existing list.

- resources — A pointer to the list of the objects associated with this Application (every item of this list is structured as [CfgObjectResource](#)). When used as an entry in [CfgDeltaApplication](#), it is a pointer to a list of resources added to the existing list. Only objects of type [CfgScript](#) can be associated with Application object through resources.

Comments

Deletion of Application(Server) X will cause the following events set out in the order of arrival:

- Modification of detect event of the alarm conditions which referred on Application X
- Modification of backupServerDBID field of the application that had Application X as a backup server
- Modification of appServerDBIDs field of all applications that had Application X in their connections
- Modification of campaign groups of all campaigns that had statServerDBID or dialerDBID fields set to Application X
- Modification of IVR objects that had IVRServerDBID field set to Application X
- Modification of solution and host objects that had SCsDBID field set to Application X
- Deletion of Application X

An application/server cannot be deleted as long as it associated with at least one client application i.e., the connection between client and server is specified within configuration object or it is included to at least one non-optional solution component, or it is assigned as DBServer to at least one table access.

Parameter options has the following structure: Each key-value pair of the primary list (TKVList *options) uses the key for the name of a configuration section, and the value for a secondary list, that also has the TKVList structure and specifies the configuration options defined within that section. Each key-value pair of the secondary list uses the key for the name of a configuration option, and the value for its current setting. Configuration options can be defined as variables of integer, character, or binary type. Names of sections must be unique within the primary list. Names of options must be unique within the secondary list.

Tip

Configuration Server is not concerned with logical meanings of application-specific configuration sections, options, or their values.

Applications of the daemon type are allowed to establish one and only one communication session to Configuration Server.

Access privileges of an application of the daemon type are determined by the access privileges of the account it is associated with. By default, a new application of the daemon type is associated with access group System (see comments to [CfgAccessGroup](#) in section Access Control Functions and Data Types). Function [ConfSetAccount](#) can be used to change the default account.

Access privileges of an application of the GUI type are determined by the access privileges of the currently logged-on person. See comments to [CfgPerson](#).

An application is allowed to establish a communication session with Configuration Server only if the currently logged-on person (for GUI applications) or the account (for daemon applications) has Execute permission with respect to this Application (see type CfgACE).

An application of CFGConfigServer type with DBID = 99 shall be pre-defined (scripted) in the Configuration Database before Configuration Server is started for the first time. The object that represents this application cannot be deleted.

An application of CFGSCE type with DBID = 100 shall be pre-defined (scripted) in the Configuration Database before Configuration Server is started for the first time. The object that represents this application cannot be deleted.

An application can be included into different solutions. Configuration Server does not provide the synchronization property tenantDBID of CfgService and property tenantDBIDs of CfgApplication. The tenantDBIDs list should be updated manually or by wizard every time a solution the application is a part of is assigned to a new tenant (the corresponding tenant's id should be added to the list). Similar (manual or by wizard) update should be made if a solution the application is included into is no longer associated with a tenant (the corresponding tenant's id should be removed from the list).

An application can not be deleted as long as there is at least one solution the application is a part of.

After upgrading from CME 5.1.x to 6.1 the following default values should be set for the application:

- workDirectory = '.'
- commandLine = name (the value of name property)
- autoRestart = CFGFalse
- startupTimeout = 90
- redundancyType=CFGHTColdStandby
- isPrimary=CFGTrue
- startupType= <should be set according to description of startupType above>

An application cannot be deleted if it has a type CFGITCUtility (53).

The name of the application can not be changed if there is, at least, one active client exist registered under this name.

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgApplication>
  <DBID value="100" />
  <name value="myName" />
  <password value="the_password" />
  <type value="19" />
  <version value="7.5" />
  <isServer value="1" />
  <state value="1" />
  <appPrototypeDBID value="100" />
  <autoRestart value="0" />
  <startupTimeout value="0" />
  <shutdownTimeout value="0" />
  <redundancyType value="0" />
  <isPrimary value="0" />
  <startupType value="3" />
</CfgApplication>
```

See Also

[CfgDeltaApplication](#)

[CfgHost](#)

CfgCallingList

Description

Calling Lists are references to tables of information about the numbers to call during an outbound campaign. These objects also specify conditions that Outbound Contact applications observe when working with these Calling Lists.

Filter Keys

| Filter Name | Type | Description |
|-----------------------|--------|---|
| dbid | int | A unique identifier of the calling list. If specified, configuration server will return information only about this calling list. |
| tenant_dbid | int | A unique identifier of the tenant. If specified, Configuration server will return information only about the calling list(s) that belong to this tenant. |
| name | string | Name of a calling list. Shall be specified as a character string. If specified, Configuration Server will return information only about the calling list(s) with that name. |
| table_access_dbid | int | A unique identifier of the table access. If specified, Configuration Server will return information only about the calling lists(s) engaged with that table access. |
| filter_dbid | int | A unique identifier of the filter. If specified, Configuration Server will return information only about the calling list(s) with that filter. |
| treatment_dbid | int | A unique identifier of the treatment. If specified, Configuration Server will return information only about the calling list(s) with that treatment. |
| log_table_access_dbid | int | A unique identifier of the log table access. If specified, Configuration Server will return |

| Filter Name | Type | Description |
|-------------|------|--|
| | | information only about the calling lists(s) engaged with that log table access. |
| script_dbid | int | A unique identifier of the calling list script. If specified, Configuration Server will return information only about the calling list(s) with that script. |
| state | int | Current state of the campaign (See CfgObjectState). If specified, Configuration Server will return information only about the campaign(s) 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 calling list is allocated. Mandatory. Once specified, cannot be changed.
- **name** — A pointer to the calling list name. Mandatory. Must be unique within the Configuration Database.
- **description** — A pointer to the calling list description.
- **tableAccessDBID** — A unique identifier of the table the calling list refers to. Reference to [CfgTableAccess](#) object of [CFGTTCallingList](#) type can be used only. Mandatory.
- **filterDBID** — A unique identifier of [CfgFilter](#) of this calling list. Optional. If specified, the filter format must have reference to the format the table access refers to.
- **treatmentDBIDs** — A pointer to list of identifiers of [CfgTreatment](#) dedicated to the calling list. Optional.
- **logTableAccessDBID** — A unique identifier of [logTableAccess](#). It is recommended to dedicate single [logTableAccessDBID](#) to all [CallingLists](#) associated with one Campaign. Reference to [CfgTableAccess](#) object of [CFGTTLogTable](#) type can be used only. Optional.
- **timeFrom** — An earliest time when dial can be done. The value is measured in seconds since 00:00:00 of current day. Default value is 08:00:00. Overwrites the settings in database table specified in field [CFGFrom](#). Mandatory.
- **timeUntil** — An latest time when dial can be done. The value is measured in seconds since 00:00:00 of current day. Default value is 18:00:00. Overwrites the settings in database table specified in field [CFGUntil](#). Mandatory.
- **maxAttempts** — A maximum number of attempts the single record can be dialed for one campaign (total). Default value is 10. Mandatory.
- **scriptDBID** — A unique identifier of the [CfgScript](#) for this calling list. Optional.
- **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

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

- modification of `callingLists` of all campaigns that included Calling List X
- deletion of Calling List X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgCallingList>
  <DBID value="101" />
  <tenantDBID value="101" />
  <name value="Calling_List" />
  <description value="A list" />
  <tableAccessDBID value="101" />
  <filterDBID value="101" />
  <logTableAccessDBID value="0" />
  <timeFrom value="28800" />
  <timeUntil value="64800" />
  <maxAttempts value="10" />
  <scriptDBID value="0" />
  <state value="1" />
</CfgCallingList>
```

See Also

[CfgDeltaCallingList](#)

[CfgTableAccess](#)

[CfgFilter](#)

[CfgCampaign](#)

CfgCallingListInfo

Description

Information about a calling list.

Attributes

- callingListDBID — A unique identifier of the [CfgCallingList](#). Mandatory.
- share — A list share factor (list weight). Default value is 10. The calling list is disabled if share factor is set to 0.
- isActive — An indicator of whether the calling list is active. See type [CfgFlag](#). Default value is CFGTrue.

CfgCampaign

Description

Campaigns are structures for organizing and managing an automated process of making outbound calls to the destinations specified in Calling Lists.

Filter Keys

| Filter Name | Type | Description |
|-------------------|--------|---|
| dbid | int | A unique identifier of the campaign. If specified, configuration server will return information only about this campaign. |
| tenant_dbid | int | A unique identifier of the tenant. If specified, Configuration server will return information only about the campaign(s) that belong to this tenant. |
| name | string | Name of a campaign. Shall be specified as a character string. If specified, Configuration Server will return information only about the campaign(s) with that name. |
| group_dbid | int | A unique identifier of group. If specified, Configuration Server will return information only about the campaign(s) with that group. |
| calling_list_dbid | int | A unique identifier of calling list. If specified, Configuration Server will return information only about the campaign(s) with that calling list. |
| script_dbid | int | A unique identifier of the campaign script. If specified, Configuration Server will return information only about the campaign(s) with that script. |
| state | int | Current state of the campaign (See CfgObjectState). If specified, Configuration Server will return information only about the campaign(s) that are |

| Filter Name | Type | Description |
|-------------|------|--------------------------|
| | | 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 campaign is allocated. Mandatory. Once specified, cannot be changed.
- **name** — A pointer to the campaign name. Mandatory.
- **description** — A pointer to the campaign description.
- **callingLists** — A pointer to list of calling lists info dedicated to campaign (every item of this list is structured as [CfgCallingListInfo](#)).
- **campaignGroups** — Not in use starting from 7.5 release. Campaign Groups are represented as [CfgCampaignGroup](#) objects under the Campaigns.
- **scriptDBID** — A unique identifier of the campaign [CfgScript](#). Optional.
- **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.

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgCampaign>
  <DBID value="101" />
  <tenantDBID value="101" />
  <name value="ACampaign" />
  <scriptDBID value="0" />
  <state value="1" />
</CfgCampaign>
```

See Also

[CfgDeltaCampaign](#)

CfgCampaignGroup

CfgCallingList

CfgCampaignGroup

Description

CfgCampaignGroup objects contain information about Campaign groups.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|--|
| dbid | int | A unique identifier of the calling list. If specified, configuration server will return information only about this campaign group. |
| tenant_dbid | int | A unique identifier of the tenant. If specified, Configuration server will return information only about the campaign group(s) that belong to this tenant. |
| name | string | Name of a calling list. Shall be specified as a character string. If specified, Configuration Server will return information only about the campaign group(s) with that name. |
| group_dbid | int | A unique identifier of group. If specified, Configuration Server will return information only about the campaign group(s) with that group. |
| state | int | Current state of the campaign group (See CfgObjectState). If specified, Configuration Server will return information only about the campaign group(s) 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.
- campaignDBID — A unique identifier of the [CfgCampaign](#) to which this campaign group is allocated.

Mandatory. Once specified, cannot be changed.

- tenantDBID — A unique identifier of the [CfgTenant](#) to which this campaign group is allocated. Read-only. Its value is populated from the associated Campaign object
- name — A pointer to the campaign group name. Mandatory. Must be unique within the Campaign.
- groupDBID — A unique identifier of the group of Agents or group of Places. Mandatory.
- groupType — A group type. Read only. See [CfgObjectType](#).
- description — A pointer to campaign group description.
- serverDBIDs — A pointer to the list of unique identifiers to [Application](#) objects. Optional. Will be used to configure connectivity to Servers associated with this Campaign. Only one Application of specific application type (CfgAppType) is allowed in the list.
- IVRProfileDBID — A unique identifier of the [CfgGVPIVRProfile](#) object. Optional.
- dialMode — A dial mode dedicated for this group. Default value is CFGDMPredict. See type [CfgDialMode](#). Mandatory.
- origDNDBID — A unique identifier of the DN where the dialing should be performed from. DNs of following types can be used to specify this parameter: CFGACDQueue and CFGRoutingPoint. Refer to [CfgDNType](#) of User Defined Variable Types. Optional.
- numOfChannels — Maximum number of outbound channels that can be used by this group at one time. Default value is 10. Mandatory.
- operationMode — An operation mode. Default value is CFGOMManual . Refer to [CfgOperationMode](#) of User Defined Variable Types. Mandatory.
- minRecBuffSize — A record buffering parameter. Default value is 4. Cannot be set to 0. Mandatory.
- optRecBuffSize — A record buffering parameter. Default value is 6. Mandatory. The value of this property must always be greater than minRecBuffSize.
- maxQueueSize — Maximal number of unprocessed Interactions submitted to Interaction Server or GVP OBN Manager. Optional.
- optMethod — An optimization method. Default value is CFGOMBusyFactor. Refer to [CfgOptimizationMethod](#) of User Defined Variable Types. Mandatory.
- optMethodValue — The value of optimization method specified by optMethod property. Refer to [CfgOptimizationMethod](#) of User Defined Variable Types for ranges and default values.

Tip

Default value for CFGOMBusyFactor is 80.

- interactionQueueDBID — A unique identifier of the [Script](#) of type CFGInteractionQueue for this campaign group. Optional.
- scriptDBID — A unique identifier of the [Script](#) for group/campaign. Optional.
- 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.

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgCampaignGroup>
  <DBID value="101" />
  <campaignDBID value="101" />
  <tenantDBID value="101" />
  <name value="CampaignGroup@CampaignGroup" />
  <groupDBID value="107" />
  <groupType value="5" />
  <IVRProfileDBID value="0" />
  <dialMode value="1" />
  <origDNDBID value="0" />
  <numOfChannels value="10" />
  <operationMode value="1" />
  <minRecBuffSize value="4" />
  <optRecBuffSize value="6" />
  <maxQueueSize value="0" />
  <optMethod value="1" />
  <optMethodValue value="80" />
  <interactionQueueDBID value="0" />
  <scriptDBID value="0" />
  <state value="1" />
</CfgCampaignGroup>
```

See Also

[CfgDeltaCampaignGroup](#)

[CfgCampaign](#)

[CfgAgentGroup](#)

[CfgPlaceGroup](#)

CfgConnInfo

Description

CfgConnInfo contains information about a connection.

Attributes

- `appServerDBID` — The unique identifier of the [Server](#) this application shall connect to as a client when it is started.
- `connProtocol` — A pointer to the name of the connection control protocol. Available values: `adp`. Default: `none`.
- `timeoutLocal` — The heart-bit polling interval measured in seconds, on client site. See comments below.
- `timeoutRemote` — The heart-bit polling interval measured in seconds, on server site. See comments below.
- `mode` — The trace mode dedicated for this connection. Refer to [CfgTraceMode](#) below. Default value: `CFGTMNoTraceMode`.
- `id` — An identifier of the server's listening port. Should correspond to `CfgPortInfo.id`.
- `transportParams` — Connection protocol's transport parameters.
- `appParams` — Connection protocol's application parameters.
- `proxyParams` — Connection protocol's proxy parameters.
- `description` — Optional description of the connection.
- `charField1` — Optional text field #1.
- `charField2` — Optional text field #2.
- `charField3` — Optional text field #3.
- `charField4` — Optional text field #4.
- `longField1` — Optional integer field #1.
- `longField2` — Optional integer field #2.
- `longField3` — Optional integer field #3.
- `longField4` — Optional integer field #4.

Comments

Tip

If client and server exchange large processing instructions, that is, packets larger than 1Mbyte, the values for `timeoutLocal` and `timeoutRemote` for this connection should not be set to less than 3 seconds. Otherwise, the connection library will be forced to disconnect the client.

CfgDN

Description

A *DN* is a communication device that is uniquely identified by a directory number, where customer interactions (for example, telephone calls or e-mails) reside and are handled.

Filter Keys

| Filter Name | Type | Description |
|---------------|--------|---|
| dbid | int | A unique identifier of a DN. If specified, Configuration Server will return information only about this DN. |
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the DNs that belong to this tenant. |
| switch_dbid | int | A unique identifier of a switch. If specified, Configuration Server will return information only about the DNs that belong to this switch. |
| dn_type | int | Type of the DN (see CfgDNType). If specified, Configuration Server will return information only about the DNs of this type. |
| place_dbid | int | A unique identifier of a place. If specified, Configuration Server will return information only about the DNs that are associated with this place. |
| no_place_dbid | int | Configuration Server will return information only about the DN(s) that) that are allowed to be assigned to the place and not associated with any place. |
| group_dbid | int | A unique identifier of the group of DNs. If specified, Configuration Server will return information only about the DNs that are associated with this group. |
| association | string | An entity associated with a DN. |

| Filter Name | Type | Description |
|-------------|--------|---|
| | | Shall be specified as a character string. If specified, Configuration Server will return information only about the DN(s) that are associated with this entity. |
| state | int | Current state of a DN (see CfgObjectState). If specified, Configuration Server will return information only about DNs that are currently in this state. |
| dn_number | string | Directory number of a DN. Shall be specified as a character string. If specified, Configuration Server will return information only about the DN(s) with that number. |
| name | string | Name of a DN. Shall be specified as a character string. If specified, Configuration Server will return information only about the DN(s) with that name. This field corresponds to the <code>alias</code> field in the advanced tab of a DN object in Configuration Manager. |
| ivr_dbid | int | A unique identifier of an IVR object (see CfgIVR). If specified, Configuration Server will return information only about the DN(s) which assigned to IVR Ports (see CfgIVRPort) of this IVR object. |

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.
- **switchDBID** — A unique identifier of the [Switch](#) to which this DN belongs. Mandatory. Once specified, cannot be changed.
- **tenantDBID** — A unique identifier of the [Tenant](#) to which this DN belongs. Read-only (set automatically according to the current value of `tenantDBID` of the switch specified in `switchDBID`). See [CfgSwitch](#).
- **type** — Type of this DN. See [CfgDNType](#). Mandatory. Once specified, cannot be changed.

Tip

DNs of the following types should be considered as routing DNs: CFGACDQueue, CFGRoutingPoint, CFGVirtACDQueue, CFGVirtRoutingPoint, CFGExtRoutingPoint, CFGServiceNumber, CFGRoutingQueue, CFGAccessResource. (See comments.)

- number — Directory number assigned to this DN within the switch. Mandatory. Must be unique within the switch for all dn types except CFGDestinationLabel and CFGAccessResource. The uniqueness of CFGAccessResource is defined by combination of number and DN type. Once specified, cannot be changed. Please see the comment regarding the parameter DNRange in [CfgSwitch](#).
- association — A pointer to the identifier of an entity permanently associated with this DN (e.g., an IVR port number, channel name, or access number).
- destDNDBIDs — A pointer to the list of identifiers of the objects (DBIDs) to which the calls residing at this DN can be routed/diverted by default. Makes sense only if type is set to CFGRoutingPoint, CFGExtRoutingPoint, CFGServiceNumber, CFGRoutingQueue, CFGACDQueue, CFGVirtACDQueue, or CFGVirtRoutingPoint, and CFGAccessResource and shall be set to NULL for all other values of type. When used as an entry in CfgDeltaDN (see below), it is a pointer to a list of identifiers of the objects added to the existing list. The DN for which this list is specified cannot be added to this list. If DN type is CFGAccessResource the property must be presented on GUI (Config Manager) with caption “Remote Resources”.
- loginFlag — An indicator of whether a login procedure is necessary to activate the telephony object associated with this DN. Read-only (set automatically according to the current value of DNLoginID below). See [CfgFlag](#). The value should not be taken into consideration if DN type is CFGAccessResource.
- DNLoginID — A pointer to the login identifier used to activate this DN. Makes sense only if type is set to CFGACDPosition, CFGExtension, CFGEAPPort, CFGVoiceMail, or CFGMixed. For type CFGAccessResource specifies the type of the resource and must be presented on GUI (Configuration Manager) with caption “Resource Type”.
- registerAll — An indicator of whether T-Server shall register this DN within the switch. Recommended to be set to CFGDRTTrue by default. See [CfgDNRegisterFlag](#).
- groupDBID — A unique identifier of the [DN Group](#) used in number translation.
- trunks — Number of trunks associated with this DN. Makes sense only if type is set to CFGDestinationLabel.
- routeType — Type of routing that applies to this DN. See type [CfgRouteType](#).
- override — The number used as a substitute of a regular directory number in certain types of routing.

Tip

The property must have the same value as number for DNs converted from SCE 5.0.

- 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.

- name — Name of this DN required if the DN is planned to be used as a target in routing strategies. If specified, must be unique within the tenant this DN belongs to. It is strongly recommended to give names to DNs of the following types: CFGACDQueue, CFGRoutingPoint, CFGVirtualACDQueue, CFGVirtualRoutingPoint, and CFGRoutingQueue.
- useOverride — An indicator of whether the override value shall be used instead of the number or name value for accessing this DN in certain types of routing. Recommended to be set to CFGTrue by default. See [CfgFlag](#).
- switchSpecificType — An integer that corresponds to a combination of switch-specific settings for this DN. Cannot be set to a zero or negative value.
- accessNumbers — A pointer to the list of structures that specify the numbers to be dialed from different switches to get this DN. Makes sense only if type is set to CFGExtRoutingPoint and CFGAccessResource. See [CfgDNAccessNumber](#).
- siteDBID — A unique identifier of Site ([CfgFolder](#)) with which this DN is associated.
- contractDBID — A unique identifier of Cost Contract ([CfgObjectiveTable](#)) with which this DN is associated.

Comments

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

- Modification of DNDBIDs of the place that had DN X assigned
- Modification of destDNDBIDs of all DNs whose destination DNs included DN X
- Modification of routedDNDBIDs of all agent groups whose route DNs included DN X
- Modification of routedDNDBIDs of all place groups whose route DNs included DN X
- Modification of DNs of all DN groups whose DNs included DN X
- Modification of routedDNDBIDs of all DN groups whose route DNs included DN X
- Modification of [CfgTreatment](#) objects which had destDNDBID field set to DN X
- Modification of [CfgIVRPort](#) objects which had DNDBID field set to DN X
- Modification of [CfgCampaignGroup](#) which had origDNDBID field set to DN X
- Deletion of DN X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgDN>
```



```
<DBID value="102" />
<switchDBID value="102" />
<tenantDBID value="1" />
<type value="1" />
<number value="001" />
<loginFlag value="1" />
<registerAll value="2" />
<groupDBID value="0" />
<trunks value="0" />
<routeType value="1" />
<state value="1" />
<userProperties>
  <list_pair key="Property">
    <str_pair key="key" value="value" />
  </list_pair>
</userProperties>
<useOverride value="2" />
<switchSpecificType value="1" />
<siteDBID value="0" />
<contractDBID value="0" />
</CfgDN>
```

See Also

[CfgDeltaDN](#)

[CfgSwitch](#)

[CfgDNGroup](#)

[CfgPlace](#)

CfgDNAccessNumber

Description

CfgDNAccessNumber displays a list of numbers that are dialed from the specified Switch to reach this DN.

It is only applicable to the External Routing Point or Access Resource DN types.

Attributes

- switchDBID — A unique identifier of the **Switch** that initiates call. Mandatory.
- number — A set of digits to be dialed by T-Server/Switch to reach specific external routing point(this DN).

Comments

The uniqueness of the CfgDNAccessNumber is defined by the combination of the switchDBID and number properties.

CfgDNGroup

Description

You can set up *Groups* of DNs for use in network-level routing algorithms and in some types of statistics. Consult solution-specific documentation to see if you need to set up DN Groups.

When you are specifying a DN Group, remember that the DNs comprising this group must have the same telephony event model.

Filter Keys

| Filter Name | Type | Description |
|---------------|--------|--|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the DN groups that belong to this tenant. |
| dn_group_type | int | Type of a DN group (see CfgDNGroupType). If specified, Configuration Server will return information only about the DN groups of this type. |
| dn_dbid | int | A unique identifier of a DN. If specified, Configuration Server will return information only about the DN groups this DN is assigned to. |
| state | int | Current state of a DN group (see CfgObjectState). If specified, Configuration Server will return information only about DN groups that are currently in this state. |
| name | string | Name of a DN group. Shall be specified as a character string. If specified, Configuration Server will return information only about the agent group(s) with that name. |
| dbid | int | A unique identifier of a group. If specified, Configuration Server will return information only about this group. |

Attributes

- `groupInfo` — A pointer to the structure `[[CfgGroup[]` containing general information about this group. Mandatory.
- `type` — Type of this DN group. Once specified, cannot be changed. Refer to `CfgDNGroupType`.
- `DNs` — A pointer to the list of the DNs that form this group (every item of this list is structured as `CfgDNInfo`).

Comments

The name of a DN group must be unique within the tenant, but can coincide with the names of access groups, place groups and agent groups of the same tenant.

The name of a DN group cannot be changed until there is at least one DN listed in this group. See `DNs` property.

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

- Modification of `groupDBID` of all DNs that used DN Group X for number translation
- Deletion of DN Group X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgDNGroup>
  <CfgGroup>
    <DBID value="124" />
    <tenantDBID value="101" />
    <name value="SomeDNs" />
    <capacityTableDBID value="0" />
    <quotaTableDBID value="0" />
    <state value="1" />
    <capacityRuleDBID value="0" />
    <siteDBID value="0" />
    <contractDBID value="0" />
  </CfgGroup>
  <type value="3" />
  <DNs>
    <CfgGroupDN>
      <DNDBID value="250" />
      <trunks value="0" />
    </CfgGroupDN>
  </DNs>
</CfgDNGroup>
```

```
<CfgGroupDN>  
  <DNDBID value="270" />  
  <trunks value="0" />  
</CfgGroupDN>  
</DNs>  
</CfgDNGroup>
```

See Also

[CfgDeltaDNGroup](#)

[CfgDN](#)

CfgDNInfo

Description

Information about a DN.

Attributes

DNDBID — A unique identifier of the **CfgDN**. Mandatory. Once specified, cannot be changed. The same value cannot be repeated within one list.

trunks — Number of trunks associated with this DN. Cannot be set to a negative value. Makes sense only for DN groups of **CFGNetworkPorts** type.

CfgDeltaAccessGroup

Description

The changes to make to a [CfgAccessGroup](#) object.

Attributes

- `deltaGroupInfo` — A pointer to the structure containing information about changes made to this group data. Mandatory.
- `deletedMemberIDs` — A pointer to the list of Persons (every item of this list is structured as [CfgID](#)) deleted from this group.
- `addedMemberIDs` — A pointer to the list of the Persons added to this group (every item of this list is structured as [CfgID](#)).

CfgDeltaActionCode

Description

The changes to make to a [CfgActionCode](#) object.

Attributes

- `deltaActionCode` — A pointer to the structure that contains information about changes made to this action code data. Mandatory.
- `deletedSubcodes` — A pointer to the list of names of the subcodes that are no longer defined within this action code (every item of this list has variable type `char*`).
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaAgentGroup

Description

The changes to make to a [CfgAgentGroup](#) object.

Attributes

- `deltaGroupInfo` — A pointer to the structure containing information about changes made to this group data. Mandatory.
- `deletedAgentDBIDs` — A pointer to the list of identifiers of the agents deleted from this group.
- `addedAgentDBIDs` — A pointer to the list of identifiers of the agents added to this group.

CfgDeltaAgentInfo

Description

The changes to make to a [CfgAgentInfo](#) object.

Attributes

- `deletedSkillDBIDs` — A pointer to the list of identifiers of the deleted skills.
- `changedSkillLevels` — A pointer to the list of changed skill levels (every item of this list is structured as [CfgSkillLevel](#)).
- `deletedAgentLoginDBIDs` — A pointer to the list of identifiers of the agent logins that this agent can no longer use.
- `changedAgentLogins` — A pointer to the list of agent logins with changed wrap-up time (every item of this list is structured as [CfgAgentLoginInfo](#)).

CfgDeltaAgentLogin

Description

The changes to make to a [CfgAgentLogin](#) object.

Attributes

- `deltaAgentLogin` — A pointer to the structure that contains information about changes made to this agent login data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaAlarmCondition

Description

The changes to make to a [CfgAlarmCondition](#) object.

Attributes

- `deltaAlarmCondition` — A pointer to the structure that contains information about changes made to this [CfgAlarmCondition](#) data.
- `deletedReactionScriptDBIDs` — A pointer to a list of identifiers of the scripts excluded from reactions to the alarm based on this alarm condition.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.
- `deletedClearanceScriptDBIDs` — A pointer to a list of identifiers of the scripts excluded from clearance to the alarm based on this alarm condition.

CfgDeltaAppPrototype

Description

The changes to make to a [CfgAppPrototype](#) object.

Attributes

- `deltaAppPrototype` — A pointer to the structure containing information about changes made to this service data. Mandatory.
- `deletedOptions` — A pointer to the list of deleted application-specific options. The structure of this parameter is described in the comments to type [CfgAppPrototype](#). An option is deleted by specifying the name of the section that this option belongs to, and the name of the option itself with any value. A whole section is deleted by specifying the name of that section and an empty secondary list.
- `changedOptions` — A pointer to the list of application-specific options whose values have been changed. The structure of this parameter is described in the comments to type [CfgAppPrototype](#). A value of an option is changed by specifying the name of the section that this option belongs to, the name of the option itself, and the new value of that option.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

Comments

Changes made to the options of an application prototype will not affect application-specific options of the existing applications associated with this prototype.

CfgDeltaApplication

Description

The changes to make to a [CfgApplication](#) object.

Attributes

- `deltaApplication` — A pointer to the structure containing information about changes made to this application data. Mandatory.
- `deletedAppServerDBIDs` — A pointer to the list of identifiers of the servers this application can no longer be a client to.
- `deletedTenantDBIDs` — A pointer to the list of identifiers of the tenants that are no longer associated with this application. For applications of `CFGTSERVER` type, cannot be specified as long as the application is associated with a switch (see [CfgSwitch](#)).
- `deletedOptions` — A pointer to the list of deleted application-specific options. The structure of this parameter is described in the comments to type [CfgApplication](#). An option is deleted by specifying the name of the section that this option belongs to, and the name of the option itself with any value. A whole section is deleted by specifying the name of that section and an empty secondary list.
- `changedOptions` — A pointer to the list of application-specific options whose values have been changed. The structure of this parameter is described in the comments to type [CfgApplication](#). A value of an option is changed by specifying the name of the section that this option belongs to, the name of the option itself, and the new value of that option.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.
- `deletedFlexibleProperties` — A pointer to the list of deleted options within `flexibleProperties` property.
- `changedFlexibleProperties` — Should not be used. The structure of the `flexibleProperties` implies only add and delete actions.
- `changedAppServerDBIDs` — A pointer to the list of structures [CfgConnInfo](#) type. Each structure contains `appServerDBID` and the information about connection parameters that have been changed.
- `deletedResources` — A pointer to the list of deleted resources (every item of this list is structured as [CfgID](#)).
- `changedResources` — A pointer to the list of structures [CfgObjectResource](#) type. Each structure contains information about the resource parameters that have been changed.
- `deletedPortInfos` — A pointer to the list of deleted resources (every item of this list is a pointer to a string containing identifier (`id`) of the listening port structure).

- `changedPortInfos` — A pointer to the list of structures `CfgPortInfo` type. Each structure contains information about the listening port parameters that have been changed.

CfgDeltaCallingList

Description

CfgDeltaCallingList is applicable for Configuration Library/Server release 5.1.5xx and later.

Attributes

- deltaCallingList — A pointer to the structure that contains information about changes made to this [CfgCallingList](#) data.
- deletedTreatmentDBIDs — A pointer to the list of deleted treatments.
- deletedUserProperties — A pointer to the list of deleted user-defined properties. Has the same structure as parameter userProperties.
- changedUserProperties — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter userProperties.

CfgDeltaCampaign

Description

CfgDeltaCampaign is applicable for Configuration Library/Server release 5.1.5xx and later.

Attributes

- `deltaCampaign` — A pointer to the structure that contains information about changes made to this `CfgCampaign` data.
- `deletedCallingListDBIDs` — A pointer to the list of deleted calling list DBID(s).
- `changedCallingLists` — A pointer to the list of changed calling lists (every item of this list is structured as `CfgCallingListInfo`).
- `deletedCampaignGroupDBIDs` — Not in use starting from release 7.5
- `changedCampaignGroups` — Not in use starting from release 7.5
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaCampaignGroup

Description

CfgDeltaCampaignGroup is applicable for Configuration Library/Server release 7.5 and later.

Attributes

- `deltaCampaignGroup` — A pointer to the structure that contains information about changes made to this `CfgCampaignGroup` data.
- `deletedServerDBIDs` — A pointer to the list of deleted applications.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaDN

Description

The changes to make to a [CfgDN](#) object.

Attributes

- `deltaDN` — A pointer to the structure [CfgDN](#) that contains information about changes made to this DN data. Mandatory.
- `deletedDestDNDBIDs` — A pointer to the list of identifiers of the objects to which the calls residing at this DN can no longer be routed by default.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.
- `deletedAccessNumbers` — A pointer to the list of deleted access numbers (every item of this list is structured as [CfgDNAccessNumber](#)).

CfgDeltaDNGroup

Description

The changes to make to a [CfgDNGroup](#) object.

Attributes

- `deltaGroupInfo` — A pointer to the structure containing information about changes made to this group data. Mandatory.
- `deletedDNDBIDs` — A pointer to the list of identifiers of the DNs deleted from this group.
- `addedDNs` — A pointer to the list of the DNs added to this group (every item of this list is structured as [CfgDNInfo](#)).
- `changedDNs` — A pointer to the list of the DNs with changed numbers of trunks (every item of this list is structured as [CfgDNInfo](#)).

CfgDeltaEnumerator

Description

The changes to make to a [CfgEnumerator](#) object.

Attributes

- `deltaEnumerator` — A pointer to the structure that contains information about changes made to [CfgEnumerator](#) data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaEnumeratorValue

Description

The changes to make to a [CfgEnumeratorValue](#) object.

Attributes

- `deltaEnumeratorValue` — A pointer to the structure that contains information about changes made to [CfgEnumeratorValue](#) data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaField

Description

The `CfgDeltaField` is applicable for Configuration Library/Server release 5.1.5xx and later.

Attributes

- `deltaField` — A pointer to the structure `CfgField` that contains information about changes made to this Field data.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaFilter

Description

`CfgDeltaFilter` is applicable for Configuration Library/Server release 5.1.5xx and later.

Attributes

- `deltaFilter` — A pointer to the structure that contains information about changes made to this [CfgFilter](#) data.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaFolder

Description

The changes to make to a [CfgFolder](#) object.

Attributes

- `deltaFolder` — A pointer to the structure that contains information about changes made to the [CfgFolder](#) data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.
- `deletedObjectIDs` — A pointer to a list of subordinate objects excluded from the folder's container list. Contains structures of type [CfgID](#).
- `deletedResources` — A pointer to the list of deleted resources (every item of this list is structured as [CfgID](#)).
- `changedResources` — A pointer to the list of structures [CfgObjectResource](#) type. Each structure contains information about the resource parameters that have been changed.

CfgDeltaFormat

Description

The CfgDeltaFormat is applicable for Configuration Library/Server release 5.1.5xx and later.

Attributes

- deltaFormat — A pointer to the structure that contains information about changes made to this **CfgFormat** data.
- deletedFieldDBIDs — A pointer to the list of deleted fields.
- deletedUserProperties — A pointer to the list of deleted user-defined properties. Has the same structure as parameter userProperties.
- changedUserProperties — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter userProperties.

CfgDeltaGVPCustomer

Description

The changes to make to a [CfgGVPCustomer](#) object.

Attributes

- `deltaGVPCustomer` — A pointer to the structure that contains information about changes made to the [CfgGVPCustomer](#) data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaGVPIVRProfile

Description

The changes to make to a [CfgGVPIVRProfile](#) object.

Attributes

- `deltaGVPIVRProfile` — A pointer to the structure that contains information about changes made to the [CfgGVPIVRProfile](#) data. Mandatory.
- `deletedServerDBIDs` — A pointer to the list of deleted DNs of type CFGGVPDID.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaGVPreSeller

Description

The changes to make to a [CfgGVPreSeller](#) object.

Attributes

- `deltaGVPreSeller` — A pointer to the structure that contains information about changes made to the [CfgGVPreSeller](#) data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaGroup

Description

The changes to make to a [CfgGroup](#) object.

Attributes

- `deltaGroup` — A pointer to the structure containing information about changes made to this group data. Mandatory.
- `deletedManagerDBIDs` — A pointer to the list of identifiers of the Person objects that have been deleted from this group.
- `deletedRouteDNDBIDs` — A pointer to the list of identifiers of the telephony objects that have been deleted from this group.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaHost

Description

The changes to make to a [CfgHost](#) object.

Attributes

- `deltaHost` — A pointer to the structure that contains information about changes made to this host data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.
- `deletedResources` — A pointer to the list of deleted resources (every item of this list is structured as [CfgID](#)).
- `changedResources` — A pointer to the list of structures [CfgObjectResource](#) type. Each structure contains information about the resource parameters that have been changed.

CfgDeltaIVR

Description

The changes to make to a [CfgIVR](#) object.

Attributes

- `deltaIVR` — A pointer to the structure containing information about changes made to this [CfgIVR](#) object. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaIVRPort

Description

The changes to make to a [CfgIVRPort](#) object.

Attributes

- `deltaIVRPort` — A pointer to the structure that contains information about changes made to this [CfgIVRPort](#) data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaObjectiveTable

Description

The changes to make to a [CfgObjectiveTable](#) object.

Attributes

- `deltaObjectiveTable` — A pointer to the structure that contains information about changes made to the [CfgObjectiveTable](#) data. Mandatory.
- `deletedObjectiveRecords` — A pointer to a list of objective records excluded from the objective table.
- `changedObjectiveRecords` — A pointer to a list of objective records changed within the objective table.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaPerson

Description

The changes to make to a [CfgPerson](#) object.

Attributes

- `deltaPerson` — A pointer to structure [CfgPerson](#) that contains information about changes made to this person's general data. Mandatory.
- `deltaAgentInfo` — A pointer to the structure containing changes made to the agent-specific information of the person. Makes sense only if the person is an agent and shall be set to NULL otherwise. See structure [CfgDeltaAgentInfo](#).
- `deletedAppRanks` — A pointer to the list of deleted application types (every item of this list has variable type [CfgAppType](#)).
- `changedAppRanks` — A pointer to the list of changed application ranks (every item of this list is structured as [CfgAppRank](#)).
- `deletedAssignedTenantDBIDs` — Not in use.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaPhysicalSwitch

Description

The changes to make to a [CfgPhysicalSwitch](#) object.

Attributes

- `deltaPhysicalSwitch` — A pointer to the structure that contains information about changes made to this [CfgPhysicalSwitch](#) data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaPlace

Description

The changes to make to a [CfgPlace](#) object.

Attributes

- `deltaPlace` — A pointer to the structure containing information about changes made to this [CfgPlace](#) data. Mandatory.
- `deletedDNDBIDs` — A pointer to the list of identifiers of the telephony objects that have been removed from this place.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaPlaceGroup

Description

The changes to make to a [CfgPlaceGroup](#) object.

Attributes

- `deltaGroupInfo` — A pointer to the structure containing information about changes made to this group data. Mandatory.
- `deletedPlaceDBIDs` — A pointer to the list of identifiers of the places deleted from this group.
- `addedPlaceDBIDs` — A pointer to the list of identifiers of the places added to this group.

Tip

Configuration Server does not place any restrictions regarding the types of DNS in the places that are added to a group.

CfgDeltaRole

Description

The changes to make to a [CfgRole](#) object.

Attributes

- `deltaRole` — A pointer to the structure containing information about changes made to this role. Mandatory.
- `deletedUserProperties` — A pointer to the list of user properties deleted from this role.
- `changedUserProperties` — A pointer to the list of user properties changed in this role.
- `deletedRoleMembers` — A pointer to the list of role members deleted from this role.
- `changedRoleMembers` — A pointer to the list of role members changed in this role.

CfgDeltaScheduledTask

Description

The changes to make to a [CfgScheduledTask](#) object.

Attributes

- `deltaScheduledTask` — A pointer to the structure that contains information about changes made to this [CfgScheduledTask](#) data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.
- `deletedResources` — A pointer to the list of deleted resources (every item of this list is structured as [CfgID](#)).
- `changedResources` — A pointer to the list of structures [CfgObjectResource](#) type. Each structure contains information about the resource parameters that have been changed.

CfgDeltaScript

Description

The changes to make to a [CfgScript](#) object.

Attributes

- `deltaScript` — A pointer to the structure that contains information about changes made to this script data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.
- `deletedResources` — A pointer to the list of deleted resources (every item of this list is structured as [CfgID](#)).
- `changedResources` — A pointer to the list of structures [CfgObjectResource](#) type. Each structure contains information about the resource parameters that have been changed.

CfgDeltaService

Description

The changes to make to a [CfgService](#) object.

Attributes

- `deltaService` — A pointer to the structure of [CfgService](#) containing information about changes made to this service data. Mandatory.
- `deletedAppServicePermissions` — Not in use.
- `changedAppServicePermissions` — Not in use.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.
- `deletedComponents` — A pointer to a list of structures of type [CfgSolutionComponent](#) that contain information about solution components that are no longer defined within this solution.
- `changedComponents` — A pointer to a list of structures of type [CfgSolutionComponent](#) that contain information about solution components within this solution whose parameters have been changed.
- `deletedComponentDefinitions` — A pointer to a list of structures of type [CfgSolutionComponentDefinition](#) that contain information about solution component definition that are no longer defined within this solution.
- `changedComponentDefinitions` — A pointer to a list of structures of type [CfgSolutionComponentDefinition](#) that contain information about solution component definition within this solution whose parameters have been changed.

CfgDeltaSkill

Description

The changes to make to a [CfgSkill](#) object.

Attributes

- `deltaSkill` — A pointer to the structure that contains information about changes made to this skill data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaStatDay

Description

The changes to make to a [CfgStatDay](#) object.

Attributes

- `deltaStatDay` — A pointer to the structure that contains information about changes made to this stat day data. Mandatory.
- `deletedStatIntervals` — A pointer to the list of numbers of stat intervals that are no longer defined within this stat day (every item of this list has variable type `int`).
- `changedStatIntervals` — A pointer to the list of structures of type [CfgStatInterval](#) that contain information about stat intervals whose values have been changed.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userPropertiesInsert text`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaStatTable

Description

The changes to make to a [CfgStatTable](#) object.

Attributes

- `deltaStatTable` — A pointer to the structure that contains information about changes made to this stat table data. Mandatory.
- `deletedStatDayDBIDs` — A pointer to the list of identifiers of stat days that are no longer defined within this table.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaSwitch

Description

The changes to make to a [CfgSwitch](#) object.

Attributes

- `deltaSwitch` — A pointer to the structure [CfgSwitch](#) that contains information about changes made to this switch data. Mandatory.
- `deletedSwitchAccessCodes` — A pointer to the list of deleted switch access codes (every item of this list is structured as [CfgSwitchAccessCode](#)).
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaTableAccess

Description

The [CfgDeltaTableAccess](#) is applicable for Configuration Library/Server release 5.1.5xx and later.

Attributes

- `deltaTableAccess` — A pointer to the structure that contains information about changes made to this [CfgTableAccess](#) data.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaTenant

Description

The changes to make to a [CfgTenant](#) object.

Attributes

- `deltaTenant` — A pointer to the structure of type [CfgTenant](#) containing information about changes made to this tenant data. Mandatory.
- `deletedServiceDBIDs` — Not in use.
- `changedServiceInfo` — Not in use.
- `deletedTenantDBIDs` — Not in use.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaTimeZone

Description

CfgDeltaTimeZone is applicable for Configuration Library/Server release 5.1.5xx and later.

Attributes

- `deltaTimeZone` — A pointer to the structure that contains information about changes made to this `CfgTimeZone`.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaTransaction

Description

The changes to make to a [CfgTransaction](#) object.

Attributes

deltaTransaction — A pointer to the structure that contains information about changes made to this [CfgTransaction](#) data. Mandatory.

deletedUserProperties — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.

changedUserProperties — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaTreatment

Description

CfgDeltaTreatment is applicable for Configuration Library/Server release 5.1.5xx and later.

Attributes

- `deltaTreatment` — A pointer to the structure that contains information about changes made to this `CfgTreatment` data.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgDeltaVoicePrompt

Description

The changes to make to a [CfgVoicePrompt](#) object.

Attributes

- `deltaVoicePrompt` — A pointer to the structure that contains information about changes made to this [CfgVoicePrompt](#) data. Mandatory.
- `deletedUserProperties` — A pointer to the list of deleted user-defined properties. Has the same structure as parameter `userProperties`.
- `changedUserProperties` — A pointer to the list of user-defined properties whose values have been changed. Has the same structure as parameter `userProperties`.

CfgEnumerator

Description

CfgEnumerator objects are available enumerations, such as *MediaType*, *Service Type*, and others.

Filter Keys

| Filter Name | Type | Description |
|-----------------|--------|---|
| dbid | int | A unique identifier of an enumerator. If specified, Configuration Server will return information only about this enumerator. |
| name | string | System name of an enumerator. Shall be specified as a character string. If specified, Configuration Server will return information only about the enumerator(s) with that name. |
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the enumerators that belong to this tenant. |
| state | int | Current state of an enumerator (see CfgObjectState). If specified, Configuration Server will return information only about enumerators that are currently in this state. |
| enumerator_type | int | A unique identifier of an enumerator type. If specified, Configuration Server will return information only about enumerators of this type. |
| display_name | string | Display name of an enumerator. Shall be specified as a character string. If specified, Configuration Server will return information only about the enumerator(s) with that display name. |

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 enumerator is allocated. Mandatory. Once specified, cannot be changed.
- **name** — A pointer to the system name of the enumerator to be used primarily by Genesys applications. Mandatory. Must be unique within the tenant environment. Once specified, can not be changed.
- **description** — A pointer to the description of the enumerator.
- **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.
- **type** — A type of the enumerator that enumerator belongs to. See [CfgEnumeratorType](#).
- **displayName** — A pointer to the display name of the enumerator to be shown on GUI and in reports. Mandatory.

Comments

The table below provides a description of predefined [CfgEnumerator](#) objects (some fields are omitted for simplicity):

| DBID | Tenant | Name | Display name | Type | Description |
|------|--------|-----------------------------|-----------------------------|------|---|
| 101 | 1(101) | MediaType | Media Type | 1 | Defines all available Media Types |
| 102 | 1(101) | ServiceType | Service Type | 1 | Defines all available Service Types |
| 103 | 1(101) | CustomerSegment | Customer Segment | 1 | Defines all available Customer Segments |
| 104 | 1(101) | IVR Text To Speech Used | IVR Text To Speech Used | 1 | Defines IVR Text To Speech Used |
| 105 | 1(101) | IVR Speech Recognition Used | IVR Speech Recognition Used | 1 | Defines IVR Speech Recognition Used |
| 106 | 1(101) | IVR Application Name | IVR Application Name | 1 | Defines IVR Application Name |

| DBID | Tenant | Name | Display name | Type | Description |
|------|--------|-----------------------------|-----------------------------|------|---|
| 107 | 1(101) | IVR Technical Result | IVR Technical Result | 1 | Defines IVR Technical Result |
| 108 | 1(101) | IVR Technical Result Reason | IVR Technical Result Reason | 1 | Defines IVR Technical Result Reason |
| 109 | 1(101) | Case ID | Case ID | 1 | Defines Case ID |
| 110 | 1(101) | Business Result | Business Result | 1 | Defines Business Result |
| 111 | 1(101) | Root Interaction ID | Root Interaction ID | 1 | Root Interaction ID |
| 112 | 1(101) | InteractionType | Interaction Type | 1 | Predefined list of interaction types supported by contact center |
| 113 | 1(101) | InteractionSubtype | Interaction Subtype | 1 | Predefined list of interaction subtypes supported by contact center |
| 114 | 1(101) | CategoryStructure | Category Structure | 4 | Customer defined |
| 115 | 1(101) | ScreeningRules | Screening Rules | 4 | Customer defined |
| 116 | 1(101) | EmailAccounts | Email Accounts | 4 | Customer defined |
| 117 | 1(101) | StopProcessingReason | StopProcessing Reason | 1 | Predefined, extended by customer |
| 118 | 1(101) | Language | Language | 1 | Extended by customer |
| 119 | 1(101) | DispositionCode | Disposition Code | 1 | Customer defined |
| 120 | 1(101) | ReasonCode | Reason Code | 1 | Customer defined |
| 121 | 1(101) | InteractionAttributes | Interaction Attributes | 1 | List of predefined interaction attributes |
| 122 | 1(101) | ContactAttributes | Contact Attributes | 1 | List of predefined contact attributes |

In the GUI representation, the name "Enumerator" was replaced with "Business Attribute".

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

- Deletion of all Enumerator values of Enumerator X
- Deletion of all folders that had Enumerator X defined as the parent object
- Deletion of Enumerator X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgEnumerator>
  <DBID value="101" />
  <tenantDBID value="1" />
  <name value="AnEnumerator" />
  <description value="Description" />
  <state value="1" />
  <type value="1" />
  <displayName value="An Enumerator" />
</CfgEnumerator>
```

See Also

[CfgDeltaEnumerator](#)

[CfgEnumeratorValue](#)

CfgEnumeratorValue

Description

CfgEnumeratorValue objects contain all of the values for all available enumerators.

Filter Keys

| Filter Name | Type | Description |
|-----------------|--------|---|
| enumerator_dbid | int | A unique identifier of an enumerator. If specified, Configuration Server will return information only about enumerator values for this enumerator. |
| default_value | int | A default value of an enumerator. If specified, Configuration Server will return information only about enumerator values with this flag set to CFGTrue. |
| dbid | int | A unique identifier of an enumerator value. If specified, Configuration Server will return information only about this enumerator value. |
| name | string | System name of an enumerator value. Shall be specified as a character string. If specified, Configuration Server will return information only about the enumerator value(s) with that name. |
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the enumerator values that belong to this tenant. |
| state | int | Current state of a enumerator value(see CfgObjectState). If specified, Configuration Server will return information only about enumerator values that are currently in this state. |
| display_name | string | Display name of an enumerator |

| Filter Name | Type | Description |
|-------------|------|--|
| | | value. Shall be specified as a character string. If specified, Configuration Server will return information only about the enumerator(s) with that display name. |

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.
- **enumeratorDBID** — A unique identifier of the [CfgEnumerator](#) to which this value is allocated. Mandatory. Once specified, cannot be changed.
- **tenantDBID** — A unique identifier of the [CfgTenant](#) to which this enumerator value is allocated. Mandatory. Once specified, cannot be changed.
- **name** — A pointer to the system name of the enumerator to be used primarily by Genesys applications. Mandatory. Must be unique within the enumerator object. Once specified, can not be changed.
- **description** — A pointer to the description of the enumerator value.
- **isDefault** — An indicator whether the value is default. See [CfgFlag](#).
- **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.
- **displayName** — A pointer to the display name of the enumerator to be shown on GUI and in reports. Mandatory.

Comments

The Following table specifies predefined [CfgEnumeratorValue](#) objects (some fields are omitted for simplicity) and ranges for each Enumerator:

| DBID | Enumerator | Name | Display name | Is default? | Description |
|------|------------|-------|--------------|-------------|---------------------|
| 100 | 101 | Voice | voice | No | Media Voice |
| 101 | 101 | Voip | voip | No | Media Voice over IP |
| 102 | 101 | Email | email | No | Media EMail |
| 103 | 101 | Vmail | vmail | No | Media Voice Mail |
| 104 | 101 | Smail | smail | No | Media Scanned |

| DBID | Enumerator | Name | Display name | Is default? | Description |
|------|------------|------------|--------------|-------------|--|
| | | | | | Mail |
| 105 | 101 | Chat | chat | No | Media Chat |
| 106 | 101 | Video | video | No | Media Video |
| 107 | 101 | Cobrowsing | cobrowsing | No | Media Cobrowsing |
| 108 | 101 | Whiteboard | whiteboard | No | Media Whiteboard |
| 109 | 101 | Appsharing | appsharing | No | Media Application Sharing |
| 110 | 101 | Webform | webform | No | Media Web Form |
| 111 | 101 | Workitem | workitem | No | Media Workitem |
| 112 | 101 | Callback | callback | No | Media Callback |
| 113 | 101 | Fax | fax | No | Media Fax |
| 114 | 101 | Imchat | imchat | No | Media IMChat |
| 115 | 101 | Busevent | busevent | No | Media Business Event |
| 116 | 101 | Alert | alert | No | Media Alert |
| 117 | 101 | Sms | sms | No | Media SMS |
| 118 | 101 | Any | any | Yes | Media Any |
| 119 | 101 | Auxwork | auxwork | No | Media AuxWork |
| 200 | 102 | Default | default | Yes | Service Type Default |
| 300 | 103 | Default | default | Yes | Customer Segment Default |
| 350 | 104 | Unknown | unknown | Yes | IVR Text To Speech Used Default |
| 353 | 104 | Unknown | unknown | Yes | IVR Speech Recognition Used Default |
| 400 | 112 | Inbound | Inbound | No | Interactions received by contact center from outside clients |
| 401 | 112 | Outbound | Outbound | No | Interactions sent from contact center to external clients |
| 402 | 112 | Internal | Internal | No | Internal |

| DBID | Enumerator | Name | Display name | Is default? | Description |
|------|------------|-----------------------------|-------------------------------|-------------|--|
| | | | | | interactions between contact center correspondents |
| 403 | 113 | InboundNew | Inbound New | No | New incoming email. Starts a new thread |
| 404 | 113 | InboundCustomerReply | Customer Reply | No | Reply from a customer |
| 405 | 113 | InboundCollaborationReply | Inbound Collaboration Reply | No | Reply from an external source |
| 406 | 113 | InboundNDR | NDR | No | Error message sent back to the system by SMTP server chain |
| 407 | 113 | OutboundNew | Outbound New | No | New message from Contact Center to a customer |
| 408 | 113 | OutboundReply | Reply | No | Reply from Contact Center to a customer |
| 409 | 113 | OutboundAcknowledgement | Acknowledgement | No | Ack message sent to customer by Contact Center |
| 410 | 113 | OutboundAutoResponse | Auto Response | No | Automated reply from Contact Center to a customer |
| 411 | 113 | OutboundRedirect | Redirect | No | Message redirect to an external resource. No reply expected. |
| 412 | 113 | OutboundCollaborationInvite | Outbound Collaboration Invite | No | New message to an external resource |
| 413 | 113 | InternalCollaborationInvite | Internal Collaboration Invite | No | New message to an internal resource |
| 414 | 113 | InternalCollaborationReply | Internal Collaboration Reply | No | Reply from an internal resource |
| 415 | 117 | Normal | Normal | No | StopProcessing Reason Normal |
| 416 | 117 | AutoResponded | Auto | No | StopProcessing |

| DBID | Enumerator | Name | Display name | Is default? | Description |
|------|------------|-----------------------|-----------------------|-------------|--|
| | | | Responded | | Reason Auto Responded |
| 417 | 117 | Terminated | Terminated | No | StopProcessing Reason Terminated |
| 418 | 117 | Sent | Sent | No | StopProcessing Reason Sent |
| 419 | 117 | Forwarded | Forwarded | No | StopProcessing Reason Forwarded |
| 420 | 117 | Re-directed | Re-directed | No | StopProcessing Reason Re-directed |
| 421 | 118 | English | English | No | Language English |
| 422 | 121 | Priority | Priority | No | Interaction Attributes Priority |
| 423 | 121 | Category | Category | No | Interaction Attributes Category |
| 424 | 121 | ServiceType | Service Type | No | Interaction Attributes Service Type |
| 425 | 121 | MediaType | Media Type | No | Interaction Attributes Media Type |
| 426 | 121 | InteractionType | Interaction Type | No | Interaction Attributes Interaction Type |
| 427 | 121 | InteractionSubtype | Interaction Subtype | No | Interaction Attributes Interaction Subtype |
| 428 | 121 | Language | Language | No | Interaction Attributes Language |
| 429 | 121 | StopProcessing-Reason | StopProcessing Reason | No | Interaction Attributes StopProcessing Reason |
| 430 | 121 | DispositionCode | Disposition Code | No | Interaction Attributes Disposition Code |
| 431 | 121 | ReasonCode | Reason Code | No | Interaction Attributes |

| DBID | Enumerator | Name | Display name | Is default? | Description |
|------|------------|-----------------|------------------|-------------|-------------------------------------|
| | | | | | Reason Code |
| 432 | 122 | FirstName | First Name | No | Contact Attributes First Name |
| 433 | 122 | LastName | Last Name | No | Contact Attributes Last Name |
| 434 | 122 | Title | Title | No | Contact Attributes Title |
| 435 | 122 | EmailAddress | Email Address | No | Contact Attributes E-mail Address |
| 436 | 122 | PhoneNumber | Phone Number | No | Contact Attributes Phone Number |
| 437 | 122 | AccountNumber | Account Number | No | Contact Attributes Account Number |
| 438 | 122 | ContactId | Contact ID | No | Contact Attributes Contact ID |
| 439 | 122 | CustomerSegment | Customer Segment | No | Contact Attributes Customer Segment |
| 440 | 122 | PIN | PIN | No | Contact Attributes PIN |

On the GUI representation, the name "EnumeratorValue" was replaced with "Attribute Value".

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgEnumeratorValue>
  <DBID value="100" />
  <enumeratorDBID value="101" />
  <tenantDBID value="1" />
  <name value="a_value" />
  <description value="Description" />
</CfgEnumeratorValue>
```

```
<isDefault value="1" />
  <state value="1" />
  <displayName value="A Value" />
</CfgEnumeratorValue>
```

See Also

[CfgDeltaEnumeratorValue](#)

[CfgEnumerator](#)

CfgField

Description

Fields are single pieces of data within more complex data structures (for example, database records).

Use Fields to define characteristics of the data in Formats. (Refer to the Outbound Contact documentation set for more information.)

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|---|
| tenant_dbid | int | A unique identifier of the tenant. If specified, Configuration Server will return information only about the fields that belong to this tenant. |
| name | string | Name of a field. Shall be specified as a character string. If specified, Configuration Server will return information only about the field(s) with that name. |
| state | int | Current state of the field (See type CfgObjectState). If specified, Configuration Server will return information only about the fields that are currently in this state. |
| type | int | A data type of field in data base(See type CfgDataType). If specified, Configuration Server will return information only about fields that have this data type. |
| field_type | int | A field type (See type CfgFieldType). If specified, Configuration Server will return information only about fields that have this field type. |
| dbid | int | A unique identifier of a field. If specified, configuration server will return information only about this field. |
| format_dbid | int | A unique identifier of format. If specified, configuration server will return information only about |

| Filter Name | Type | Description |
|-------------|------|------------------------------------|
| | | fields that belong to this format. |

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](#) that this field belongs to. Mandatory. Once specified, cannot be changed.
- **name** — A field name in data base. Due to restrictions of database engine the recommended length for property name is 1-12 characters. Mandatory, once specified cannot be changed.
- **type** — A data type of field in data base. Mandatory, once specified cannot be changed. See [CfgDataType](#).
- **description** — A pointer to the description of field. Optional.
- **length** — A length of field in data base. Optional, once specified cannot be changed.
- **fieldType** — A field type. Refer to [CfgFieldType](#) of User Defined Variable types. Mandatory, once specified cannot be changed.
- **defaultValue** — A pointer to the default value of field. Specify what value to insert when a user does not enter a value. Optional.
- **isPrimaryKey** — A flag which determines whether or not a field is used as primary key. Once specified cannot be changed. See [CfgFlag](#).
- **isUnique** — A flag which determines whether or not a field is used as unique. Once specified cannot be changed. See [CfgFlag](#).
- **isNullable** — A flag which determines whether or not a field can allow null values (NULLs) for the data in that field. Once specified cannot be changed. See [CfgFlag](#).
- **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

A field cannot be deleted as long as it is associated with at least one format (see [CfgFormat](#)).

Uniqueness of object is defined by combination of `name` and `fieldType` properties.

The table below lists the outbound mandatory fields and their default settings. Fields should be created as default objects in Configuration Server under the `Environment` folder.

| fieldType | name | description | isPrimary Key | isUnique | isNullable | default Value | type |
|-----------------------|---------------------|--|---------------|----------|------------|---|---------------|
| CFGFTRecordID | Record_id | Unique record identifier | No | Yes | No | No | INT |
| CFGFTPhone | contact_info | Contact Info | No | No | No | No | VARCHAR (128) |
| CFGFTPhoneType | contact_info_type | Contact Info Type | No | No | No | 1= GctiCtTyHomePhone from GctiContactType of Gcti.h | INT |
| CFGFTRecordType | Record_type | Record type | No | No | No | 2= GctiRecTyGeneral from GctiRecordType of Gcti.h | INT |
| CFGFTRecordStatus | Record_status | Record status | No | No | No | 1= GctiRecStReady from GctiRecordStatus of Gcti.h | INT |
| CFGFTDialResult | Call_result | Dial result | No | No | Yes | 28 = GctiCSUnknown from GctiCallState of Gcti.h | INT |
| CFGFTNumberOfAttempts | OfAttempts | Number of attempts has been made, excluding re-dials in case of errors | No | No | No | 0 | INT |
| CFGFTScheduledTime | Call_scheduled_time | Time, when scheduled call must be done, seconds since midnight of 01/01/1970 | No | No | Yes | No | INT |
| CFGFTCallTime | Call_time | Time when last call or dial attempt has been done, seconds since | No | No | Yes | No | INT |

| fieldType | name | description | isPrimary Key | isUnique | isNullable | default Value | type |
|--------------------|----------------|---|---------------|----------|------------|----------------|--------------|
| | | midnight of 01/01/1970 | | | | | |
| CFGFTFrom | daily_from | Earliest time to perform the call. Seconds from midnight. | No | No | No | 28800 = 8AM | INT |
| CFGFTUntil | daily_till | Latest time to perform the call. Seconds from midnight | No | No | No | 64800 = 6PM | INT |
| CFGFTTimeZone | time_zone_dbid | Time zone. DBID from Configuration Data Base. | No | No | No | 122="PST" DBID | INT |
| CFGFTCampaign | campaign_id | DBID of the campaign with respect to the last dial attempt has been made. | No | No | Yes | No | INT |
| CFGFTAgentID | agent_id | Agent login identifier | No | No | Yes | No | VARCHAR (32) |
| CFGFTChainID | chain_id | Unique identifier of chain | Yes | No | No | No | INT |
| CFGFTNumberInChain | num_in_chain | Unique identifier of record within chain | Yes | No | No | No | INT |
| CFGFTGroupDBID | group_id | AgentGroup or PlaceGroup unique identifier (DBID) | No | No | Yes | No | INT |
| CFGFTAppDBID | app_id | Application unique | No | No | Yes | No | INT |

| fieldType | name | description | isPrimary Key | isUnique | isNullable | default Value | type |
|---------------------|-------------|---|---------------|----------|------------|---------------|--------------|
| | | identifier(DBID) | | | | | |
| CFGFTTreatment | history | Treatments History | No | No | Yes | No | VARCHAR(255) |
| CFGFTMediaReference | ref | Reference to media body to be sent in case of treatment | No | No | Yes | No | INT |
| CFGFTEmailSubject | subject | Email Subject | No | No | Yes | No | VARCHAR(255) |
| CFGFTEmailTemplate | template_id | Email Template ID | No | No | Yes | No | INT |
| CFGFTSwitchID | Switch_id | Switch ID | No | No | Yes | No | INT |

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgField>
  <DBID value="101" />
  <tenantDBID value="1" />
  <name value="AField" />
  <type value="1" />
  <description value="Description" />
  <length value="0" />
  <fieldType value="1" />
  <isPrimaryKey value="1" />
  <isUnique value="2" />
  <isNullable value="1" />
  <state value="1" />
</CfgField>
```

See Also

[CfgDeltaField](#)

[CfgFormat](#)

CfgFilter

Description

Use *Filters* to specify conditions for data selection from a data repository (for example, from a database table).

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|--|
| dbid | int | A unique identifier of the filter. If specified, configuration server will return information only about this filter. |
| tenant_dbid | int | A unique identifier of the tenant. If specified, Configuration server will return information only about the filter(s) that belong to this tenant. |
| name | string | Name of a filter. Shall be specified as a character string. If specified, Configuration Server will return information only about the filter(s) with that name. |
| format_dbid | int | A unique identifier of format this filter relates to. If specified, Configuration Server will return information only about the filter(s) with that format. |
| state | int | Current state of the filter (See CfgObjectState). If specified, Configuration Server will return information only about the filter(s) 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 filter is allocated. Mandatory. Once

specified, cannot be changed.

- name — A pointer to filter name. Must be unique within tenant environment. Mandatory.
- description — A pointer to filter description.
- formatDBID — A unique identifier of [CfgFormat](#) to which this filter is dedicated. Once specified cannot be changed. Mandatory.
- 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. Among other properties must include kv-pair that specifies the filter sequence which application can use to read information from data table having specified format.

Comments

Filter used by Outbound applications is defined by two kv-pairs that must be present in `userProperties` under `default` section:

Two ways of editing are provided:

- Automated. The user has a list of fields and is able to create expressions using GUI-like any SQL wizard. The only simple expressions are created i.e., list of `<field>`, `<sign>`, and `<value>` separated by AND and OR
- Manual (Advanced). The user types the statement manually and is totally responsible for the correct syntax.

If value of kv-pair exceeds character string limit (>255 symbols) the string has to be separated into two or more strings. Two or more kv-pairs with same names have to be used to locate such string in `userProperties`.

The order of kv-pairs in `userProperties` is defined by application the filter is intended to. String concatenation is application responsibility.

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

- Modification of `filterDBID` of all calling lists that refer on the Filter X
- Deletion of Filter X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgFilter>
  <DBID value="101" />
  <tenantDBID value="101" />
  <name value="Test_Filter" />
  <formatDBID value="104" />
  <state value="1" />
</CfgFilter>
```

See Also

[CfgDeltaFilter](#)

[CfgTableAccess](#)

[CfgFormat](#)

CfgFolder

Description

Folders can be used to create hierarchies of other types of objects. Each folder can contain a collection of objects of a single type.

Filter Keys

| Filter Name | Type | Description |
|----------------|--------|--|
| dbid | int | A unique identifier of a folder. If specified, Configuration Server will return information only about this folder. |
| name | string | Name of a folder. Shall be specified as a character string. If specified, Configuration Server will return information only about the folder(s) with that name. |
| owner_dbid | int | A unique identifier of an owner object. If specified, Configuration Server will return information only about the folders that belong to this object. Must be used in conjunction with the owner_type filter. |
| owner_type | int | A type of an owner object. Must be used in conjunction with the owner_dbid filter. |
| type | int | A type of a folder. If specified, Configuration Server will return information only about folders of this type. |
| default_folder | int | A flag which selects among the folders belonging to some owner object the topmost one, e.g. that which does not have a parent folder above. Must be used in conjunction with owner_type and owner_dbid filters. Most likely will be used with type filter. |
| object_dbid | int | A unique identifier of a subordinate object. If specified, Configuration Server will return |

| Filter Name | Type | Description |
|---------------------------|------|--|
| | | information only about the folder that contains this object. Must be used in conjunction with the <code>object_type</code> filter. |
| <code>object_type</code> | int | A type of a subordinate object. Must be used in conjunction with the <code>object_dbid</code> filter. |
| <code>state</code> | int | Current state of a folder (see type <code>CfgObjectState</code>). If specified, Configuration Server will return information only about folders that are currently in this state. |
| <code>folder_class</code> | int | A class of a folder. If specified, Configuration Server will return information only about folders of this class. |

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.
- `name` — A pointer to name of the folder. Mandatory. Must be unique within the parent object.
- `type` — Type of the objects this folder may contain. A folder may contain either objects of this type or subfolders with the same value of type property. See the `CfgObjectType` enumeration.
- `ownerID` — A structure containing the object type and DBID of the folder's owner object. Unlike `parentID`, this field defines the folder's logical affiliation rather than its hierarchical affiliation. An owner may be an object of the following types:
 - `CfgTenant`
 - `CfgSwitch`
 - `CfgIVR`
 - `CfgEnumerator`.
 See `CfgID`.
- `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.
- `objectIDs` — Pointer to the list of `CfgID` objects containing the type and DBID of the objects subordinate to this folder. Only objects of the type equal to the folder's type property or subfolders of this type may be contained in this list.
- `parentID` — A structure containing object type and DBID of the folder's parent, e.g. object which stands higher in the hierarchy. This may be another folder, if this folder is a subfolder, or this field may

coincide with ownerID field if this folder is a topmost (default) one. A parent may be an object of the following types: [CfgFolder](#), [CfgTenant](#), [CfgSwitch](#), [CfgIVR](#), or [CfgEnumerator](#). See [CfgID](#).

- `folderClass` — The class of the Folder. Refer to [CfgFolderClass](#) enumeration.
- `customType` — User classifier of the Folder. Optional.
- `resources` — A pointer to the list of the objects associated with this Folder (every item of this list is structured as [CfgObjectResource](#)).
 - When used as an entry in [CfgDeltaFolder](#), it is a pointer to a list of resources added to the existing list. Only objects of the following types can be associated with Folder object through resources:
 - [CfgFolder](#)
 - [CfgObjectiveTable](#)
 - [CfgGVPIVRProfile](#)
 - [CfgGVPCustomer](#)
 - [CfgTimeZone](#)
 - [CfgHost](#)

Comments

- A folder may contain objects of the type equal to the folder's type property or subfolders of this type.
- An object may be contained in one and only one folder (has only one parent).
- A folder may be a subfolder of only one parent folder or does not have a parent folder at all (be a default folder under some parent object)
- There can not be more than one default folder of particular type for some parent object.
- A folder can not be removed as long as it has at least one subordinate object.
- A Configuration Unit is a GUI name for the folder of type [CfgFolder](#). Unlike other folders, this folder can not contain ordinary objects, but may contain folders of any type including folders of type [CfgFolder](#) (Configuration Units)
- Configuration Units can only be created under the Tenant object or other Configuration Unit.

XML Representation

Important

This XML was created using the Configuration Server 7.5 schema.

```
<CfgFolder>
  <DBID value="101" />
  <name value="A Folder" />
  <type value="3" />
</CfgFolder>
```

```
<CfgOwnerID>
  <CSID value="0" />
  <DBID value="1" />
  <type value="7" />
</CfgOwnerID>
<state value="1" />
<objectIDs>
  <CfgObjectID>
    <CSID value="0" />
    <DBID value="223" />
    <type value="22" />
  </CfgObjectID>
  <CfgObjectID>
    <CSID value="0" />
    <DBID value="99" />
    <type value="20" />
  </CfgObjectID>
</objectIDs>
<CfgParentID>
  <CSID value="0" />
  <DBID value="1" />
  <type value="7" />
</CfgParentID>
<folderClass value="1" />
<customType value="0" />
</CfgFolder>
```

See Also

[CfgDeltaFolder](#)

CfgFormat

Description

Use *Formats* to specify Fields that form a data structure (for example, a database table).

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|--|
| tenant_dbid | int | A unique identifier of the tenant. If specified, Configuration server will return information only about the formats that belong to this tenant. |
| state | int | Current state of the format (See CfgObjectState). If specified, Configuration Server will return information only about the formats that are currently in this state. |
| dbid | int | A unique identifier of a format. If specified, configuration server will return information only about this format. |
| name | string | Name of a format. Shall be specified as a character string. If specified, Configuration Server will return information only about the format(s) with that name. |
| field_dbid | int | A unique identifier of the field. If specified, Configuration Server will return information only about the format(s) which consists this field. |

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 format is allocated. Mandatory. Once specified, cannot be changed.

- name — A pointer to the name of the format. Due to restrictions of database engine the recommended length for property name is 1-12 characters. Mandatory, once specified cannot be changed.
- description — A pointer to the description of format.
- fieldDBIDs — A pointer to the list of identifiers of [CfgField](#) objects that this format consists of.
- 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 format can be dedicated to several objects of type [CfgTableAccess](#).

A format cannot be deleted as long as it is associated with at least one TableAccess or Filter. The list of fields cannot be changed as long as it is associated with at least one TableAccess or Filter. See:

- [CfgTableAccess](#)
- [CfgFilter](#)

[CfgField](#) must be unique within CfgFormat.

Predefined Formats

| Name | Tenant (if applicable) | Description | Fields |
|--------------------|------------------------|--|-----------------------|
| Default_Outbound_6 | Environment | Default format for Outbound Suite (before 7.0 release) | CFGFTRecordID |
| | | | CFGFTPhone |
| | | | CFGFTRecordType |
| | | | CFGFTRecordStatus |
| | | | CFGFTDialResult |
| | | | CFGFTNumberOfAttempts |
| | | | CFGFTScheduledTime |
| | | | CFGFTCallTime |
| | | | CFGFTFrom |
| | | | CFGFTUntil |
| | | | CFGFTTimeZone |
| | | | CFGFTCampaignID |
| | | | CFGFTAgentID |
| | | | CFGFTChainID |

| Name | Tenant (if applicable) | Description | Fields |
|------------------------|------------------------|---|-----------------------|
| | | | CFGFTNumberInChain |
| | | | CFGFTPhoneType |
| Default_Outbound_70 | Environment | Default format for Outbound Suite (starting from 7.0 release) | CFGFTRecordID |
| | | | CFGFTPhone |
| | | | CFGFTRecordType |
| | | | CFGFTRecordStatus |
| | | | CFGFTDialResult |
| | | | CFGFTNumberOfAttempts |
| | | | CFGFTScheduledTime |
| | | | CFGFTCallTime |
| | | | CFGFTFrom |
| | | | CFGFTUntil |
| | | | CFGFTTimeZone |
| | | | CFGFTCampaignID |
| | | | CFGFTAgentID |
| | | | CFGFTChainID |
| | | | CFGFTNumberInChain |
| | | | CFGFTPhoneType |
| | | | CFGFTGroupDBID |
| | | | CFGFTAppDBID |
| | | | CFGFTTreatments |
| | | | CFGFTMediaReference |
| | | | CFGFTEmailSubject |
| | | | CFGFTEmailTemplateID |
| | | | CFGFTSwitchID |
| Default_DoNotCall_List | Environment | Default format for Do Not Call List | CFGFTPhone |

XML Representation

Important

This XML was created using the Configuration Server 7.5 schema.

```
<CfgFormat>
  <DBID value="101" />
  <tenantDBID value="1" />
  <name value="MyFormat" />
  <description value="Description" />
  <fieldDBIDs>
    <DBID value="111" />
    <DBID value="112" />
  </fieldDBIDs>
  <state value="1" />
</CfgFormat>
```

See Also

[CfgDeltaFormat](#)

[CfgField](#)

[CfgTableAccess](#)

CfgGVPCustomer

Description

A *GVP Customer* is a collection of GVP IVRProfiles.

Filter Keys

| Filter Name | Type | Description |
|---------------|--------|---|
| dbid | int | A unique identifier of a folder. If specified, Configuration Server will return information only about this folder. |
| name | string | Name of a GVP Customer. Shall be specified as a character string. If specified, Configuration Server will return information only about the GVP Customers with that name. |
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the GVP Customers that belong to this tenant. |
| reseller_dbid | int | A unique identifier of a GVP Reseller. If specified, Configuration Server will return information only about the GVP Customers that belong to this GVP Reseller. |
| state | int | Current state of a GVP Customer (see type CfgObjectState). If specified, Configuration Server will return information only about GVP Customers 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.

- resellerDBID — A unique identifier of the [CfgGVPre reseller](#) to which this GVP Customer is allocated. Mandatory. Once specified, cannot be changed
- tenantDBID — A unique identifier of the [CfgTenant](#) to which this GVP Customer is allocated. Populated from GVP Reseller. Read-only
- name — A pointer to name of the GVP Customer. Mandatory. Must be unique within the Configuration Database.
- displayName — A name for this GVP Customer that appears on the console display
- channel — The GVP channel used by this GVP Customer.
- notes — Optional notes and information relevant for this GVP Customer.
- isProvisioned — If this GVP Customer is provisioned in the switch database. Refer to [CfgFlag](#).
- isAdminCustomer — If this GVP Customer is classified as an Administrative Customer. Refer to [CfgFlag](#).
- timeZoneDBID — A [CfgTimeZone](#) associated with this GVP Customer
- 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.

See Also

[CfgDeltaGVPCustomer](#)

[CfgGVPIVRProfile](#)

CfgGVPIVRProfile

Description

A *GVP IVR Profile* is the IVR Profile used by a GVP Customer to process calls to that customer. Each GVP IVR Profile is associated with one GVP Customer. GVP IVR Profiles are also associated with GVP IVR Profile Groups; with a group containing one or more profiles, and one profile being associated with one or more groups

Filter Keys

| Filter Name | Type | Description |
|---------------|--------|---|
| dbid | int | A unique identifier of a folder. If specified, Configuration Server will return information only about this folder. |
| name | string | Name of a GVP IVR Profile. Shall be specified as a character string. If specified, Configuration Server will return information only about the GVP IVR Profiles with that name. |
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the GVP IVR Profiles that belong to this tenant. |
| reseller_dbid | int | A unique identifier of a GVP Reseller. If specified, Configuration Server will return information only about the GVP IVR Profiles that belong to this GVP Reseller. |
| customer_dbid | int | A unique identifier of a GVP Customer. If specified, Configuration Server will return information only about the GVP IVR Profiles that belong to this GVP Customer. |
| state | int | Current state of a GVP IVR Profile (see type CfgObjectState). If specified, Configuration Server will return information only about GVP IVR Profiles 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.
- **customerDBID** — A unique identifier of the [CfgGVPCustomer](#) to which this GVP IVR Profile is allocated. Mandatory. Once specified, cannot be changed
- **resellerDBID** — A unique identifier of the [CfgGVPre reseller](#) to which this GVP IVR Profile is allocated. Populated from GVP Customer. Read-only
- **tenantDBID** — A unique identifier of the [CfgTenant](#) to which this GVP IVr Profile is allocated. Populated from GVP Customer. Read-only
- **name** — A pointer to name of the GVP IVR Profile. Mandatory. Must be unique within the GVP Customer.
- **displayName** — A name for this GVP IVR Profile that appears on the console display
- **type** — A type of this GVP IVR Profile. Refer to [CfgIVRProfileType](#).
- **notes** — Optional notes and information relevant for this GVP IVR Profile.
- **description** — An optional short description of this GVP IVR Profile.
- **startServiceDate** — The date when this GVP IVR Profile takes effect.
- **endServiceDate** — The date when this GVP IVR Profile ends. This date must be the same as, or later than, the Start of Service.
- **isProvisioned** — If this GVP IVR Profile is provisioned in the switch database. Refer to [CfgFlag](#).
- **tfn** — A list of toll-free prefixes which this GVP IVR Profile can access.
- **status** — The status of this GVP IVR Profile.
- **DIDDBIDs** — A pointer to the list of unique identifiers to [DN](#) objects of type CFGGVPDID. Optional.
- **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.

See Also

[CfgDeltaGVPIVRProfile](#)

CfgGVPReseller

A *GVP Reseller* is a collection of GVP Customers.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|---|
| dbid | int | A unique identifier of a folder. If specified, Configuration Server will return information only about this folder. |
| name | string | Name of a GVP Reseller. Shall be specified as a character string. If specified, Configuration Server will return information only about the GVP Resellers with that name. |
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the GVP Resellers that belong to this tenant. |
| state | int | Current state of a GVP Reseller (see type CfgObjectState). If specified, Configuration Server will return information only about GVP Resellers 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 GVP Reseller is allocated.
- **name** — A pointer to name of the GVP Reseller. Mandatory. Must be unique within the configuration database.
- **displayName** — A name for this GVP Reseller that appears on the console display
- **startDate** — The date when this GVP Reseller starts.
- **isParentNSP** — If `CFGTrue`, this GVP Reseller is a Parent Network Service Provider. Refer to [CfgFlag](#).
- **timeZoneDBID** — A [CfgTimeZone](#) associated with this GVP Reseller

- notes — Optional notes and information relevant for this GVP Reseller.
- 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.

See Also

[CfgDeltaGVPreSeller](#)

[CfgGVPCustomer](#)

CfgGroup

Description

A group.

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 **Tenant** that this group belongs to. Mandatory. Once specified, cannot be changed.
- **name** — A pointer to name of the group. Mandatory. See comments to:
 - **CfgAgentGroup**
 - **CfgPlaceGroup**
 - **CfgDNGroup**
- **managerDBIDs** — A pointer to the list of the identifiers of the **CfgPerson** objects that serve as supervisors of this group. Applicable for **CfgAgentGroup** only. When used as an entry in **CfgDeltaGroup**, it is a pointer to a list of identifiers of the Persons added to the existing list. A person assigned as a supervisor to this Group should belong either to the same Tenant as the Group or to the Tenant Environment.
- **routeDNDBIDs** — A pointer to the list of identifiers of the telephony objects from which calls can be routed/diverted to this group. This list can include **DNs** whose types are set to **CFGRoutingPoint**, **CFGExtRoutingPoint**, **CFGServiceNumber**, **CFGRoutingQueue** or **CFGACDQueue**, **CFGVirtACDQueue**, and **CFGVirtRoutingPoint**. When used as an entry in **CfgDeltaGroup**, it is a pointer to a list of identifiers of the DNs added to the existing list.
- **capacityTableDBID** — A unique identifier of the **Stat Table** of **CFGCapacityTable** type associated with this group.
- **quotaTableDBID** — A unique identifier of the **Stat Table** of **CFGQuotaTable** type associated with this group.
- **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.
- **capacityRuleDBID** — A unique identifier of the capacity rule (**Script**) associated with this group. Applicable for **CfgPlaceGroup** and **CfgAgentGroup** only.
- **siteDBID** — A unique identifier of Site (**Folder**) with which this Group is associated.

- `contractDBID` — A unique identifier of Cost Contract ([Objective Table](#)) with which this Group is associated.

Comments

For special requirements that apply to the entities of `CfgGroup` when it is used as part of an access group, see the description of parameter `groupInfo` of type [CfgAccessGroup](#) in section [Access Control Functions and Data Types](#).

See Also

[CfgDeltaGroup](#)

CfgHost

Description

Hosts are the computers that run the various CTI server applications in an environment.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|---|
| os_type | int | Type of the operating system (see CfgOSType). If specified, Configuration Server will return information only about the hosts that use operating systems of this type. |
| host_type | int | Type of the host (see CfgHostType). If specified, Configuration Server will return information only about the hosts of this type. |
| state | int | Current state of a host (see CfgObjectState). If specified, Configuration Server will return information only about hosts that are currently in this state. |
| name | string | Name of a host. Shall be specified as a character string. If specified, Configuration Server will return information only about the host(s) with that name. |
| dbid | int | A unique identifier of a host. If specified, Configuration Server will return information only about this host. |
| scs_dbid | int | A unique identifier of SCS. If specified, Configuration Server will return information only about hosts controlled by this SCS. |

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.
- **name** — A pointer to name of the host. Mandatory. Must be unique within the Configuration Database. Cannot be changed as long as at least one server is assigned to this host.
- **HWID** — Not in use.
- **IPAddress** — A pointer to the IP address of the host. Optional. Must be unique within the Configuration Database. Max length 64 symbols.
- **OSinfo** — A pointer to the structure containing information about the operating system of this host. Once specified, cannot be set to NULL. See structure [CfgOS](#).
- **type** — Type of the host. Mandatory. Once specified, cannot be changed. See type [CfgHostType](#).
- **address** — Not in use.
- **contactPersonDBID** — Not in use.
- **comment** — Not in use.
- **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.
- **LCAPort** — A port on which the Local Control Agent for this host is supposed to be running. Mandatory. Default value for migration from 5.1.xxx to 5.9.xxx is 4999, for newly created hosts the value must be specified. Allowed value is any positive whole number within 0-9999 range.

Tip

It is not recommended to change the value of `LCAPort` if either any application already connected to LCA or SCS has already been started control of the LCA.

- **SCSDBID** — A unique identifier of an [CfgApplication](#) of `CFGSCS` type which is supposed to monitor/control this host. This property is valid only if Distributed SCS functionality is enabled.
- **resources** — A pointer to the list of the objects associated with this Host (every item of this list is structured as [CfgObjectResource](#)).
When used as an entry in [CfgDeltaHost](#), it is a pointer to a list of resources added to the existing list. Only objects of the following types can be associated with Host object through resources:
 - [CfgEnumeratorValue](#)
 - [CfgScript](#)
 - [CfgApplication](#)

Comments

A host cannot be deleted as long as there is at least one server associated with it. See:

- [CfgServer](#)
- [CfgApplication](#)

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgHost>
  <DBID value="101" />
  <name value="host-name" />
  <IPAddress value="1.2.3.4" />
  <CfgOSinfo>
    <OSType value="1" />
  </CfgOSinfo>
  <type value="1" />
  <contactPersonDBID value="0" />
  <state value="1" />
  <userProperties>
    <list_pair key="Property">
      <str_pair key="key" value="value" />
      <str_pair key="key" value="value" />
    </list_pair>
  </userProperties>
  <LCAPort value="0000" />
  <SCSDBID value="104" />
</CfgHost>
```

See Also

[CfgDeltaHost](#)

[CfgApplication](#)

CfgID

Description

An object ID.

Attributes

- CSID — Reserved field.
- DBID — A unique identifier of the object.
- type — An object's type. See [CfgObjectType](#).

CfgIVR

Description

An *IVR* (Interactive Voice Response) is a telephony object that contains IVR Ports; it is controlled through IVR interface drivers.

Filter Keys

| Filter Name | Type | Description |
|-----------------|--------|--|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the DNs that belong to this tenant. |
| name | string | Name of IVR. Shall be specified as a character string. If specified, Configuration Server will return information only about the IVR with that name. |
| type | int | Type of the IVR (see type CfgIVRType). If specified, Configuration Server will return information only about the IVR of this type. |
| ivr_server_dbid | int | A unique identifier of an IVR interface server. If specified, Configuration Server will return information only about the IVRs that associated with this IVR interface server. |
| state | int | Current state of an IVR (see CfgObjectState). If specified, Configuration Server will return information only about IVRs that are currently in this state. |
| dbid | int | A unique identifier of an IVR. If specified, Configuration Server will return information only about this IVR. |

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 IVR belongs. Once specified cannot be changed.
- **name** — A pointer to the name of the IVR. Mandatory. Must be unique within configuration database.
- **description** — A pointer to the description of the IVR.
- **type** — Type of this IVR. See type [CfgIVRType](#). Mandatory. Once specified cannot be changed.
- **version** — A pointer to the version of the IVR. Mandatory.
- **IVRServerDBID** — A unique identifier of the [CfgApplication](#) of CfgIVRInterfaceServer type permanently associated to serve this IVR.
- **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

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

- Deletion of all IVR Ports of IVR X
- Deletion of all folders that had IVR X defined as the parent object
- Deletion of IVR X

See Also

[CfgDeltaIVR](#)

[CfgIVRPort](#)

CfgIVRPort

Description

An *Interactive Voice Response (IVR) Port* is a telephony object uniquely identified by the numbers within an IVR at which telephone calls may reside and be handled.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|--|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the IVR ports that belong to this tenant. |
| state | int | Current state of a IVR port (see CfgObjectState). If specified, Configuration Server will return information only about IVR ports that are currently in this state. |
| ivr_dbid | int | A unique identifier of the IVR. If specified, Configuration Server will return the information only about the IVR ports that belong to this IVR. |
| dn_dbid | int | A unique identifier of the DN. If specified, configuration Server will return the information about IVR port associated with this DN. |
| port_number | string | A port number. Shall be specified as a character string. If specified, Configuration Server will return information only about the port(s) with that number. |
| dbid | int | A unique identifier of the IVR port. If specified, Configuration Server will return information only about this IVR port. |

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 IVR port belongs. Read-only (set automatically according to the current value of tenantDBID of the IVR specified in IVRDBID). See type [CfgIVR](#).
- **portNumber** — A pointer to the string, representing number associated with channel on IVR. Mandatory. Must be unique within one IVR. Once specified cannot be changed.
- **description** — A pointer to the string describing IVR port.
- **IVRDBID** — A unique identifier of [CfgIVR](#) this IVR port belongs to. Mandatory. Once specified, can not be changed
- **DNDBID** — A unique identifier of [CfgDN](#) associated with this IVR port.
- **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

An IVR port can only be associated with one IVR.

See Also

[CfgDeltaIVRPort](#)

[CfgIVR](#)

[CfgDN](#)

CfgOS

Description

CfgOS gives information about an operating system.

Attributes

- `OStype` — Type of the operating system. See type `CfgOSType`. Mandatory.
- `OSversion` — A pointer to the version of the operating system.

CfgObjectResource

Description

CfgObjectResource contains information about object's resource and related optional parameters. The role of the resource and content of optional parameter is dictated by the value `resourceType` property. See [CfgResourceType](#).

Attributes

- `resourceType` — Type of the resource. See type [CfgResourceType](#). Mandatory.
- `objectDBID` — An identifier of the object referred by this resource. Type of the object is defined by `objectType` property
- `objectType` — Type of the object referred by this resource. See type [CfgObjectType](#). Mandatory.
- `description` — Optional description of the resource
- `charField1` — Optional text field #1
- `charField2` — Optional text field #2
- `charField3` — Optional text field #3
- `charField4` — Optional text field #4
- `longField1` — Optional integer field #1
- `longField2` — Optional integer field #2
- `longField3` — Optional integer field #3
- `longField4` — Optional integer field #4

CfgObjectiveTable

Description

Objective Tables are collections of Objective Records.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|---|
| dbid | int | A unique identifier of an objective table. If specified, Configuration Server will return information only about this objective table. |
| name | string | Name of an objective table. Shall be specified as a character string. If specified, Configuration Server will return information only about the objective table(s) with that name. |
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the objective tables that belong to this tenant. |
| state | int | Current state of an objective table (see CfgObjectState). If specified, Configuration Server will return information only about objective tables 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 objective table is allocated. Mandatory. Once specified, cannot be changed.
- name — A pointer to the name of the objective table. Mandatory. Must be unique within the tenant environment.

- description — A pointer to the description of the objective table.
- objectiveRecords — A pointer to the list of identifiers of objective records comprising this objective table.
- 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.
- prepaidCost — A flat rate for the cost per call
- timeZoneDBID — A [CfgTimeZone](#) associated with this Cost Contract
- timeStart — The date on which to start the Volume Rate Contract.
- timeEnd — The date on which to end the Volume Rate Contract.
- type — Objective Table type. Refer to [CfgObjectiveTableType](#).

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgObjectiveTable>
  <DBID value="1002" />
  <tenantDBID value="101" />
  <name value="XYZCostContract" />
  <objectiveRecords>
    <CfgObjectiveTableRecord>
      <mediaTypeDBID value="1001" />
      <serviceTypeDBID value="1023" />
      <customerSegmentDBID value="1024" />
      <objectiveThreshold value="0" />
      <objectiveDelta value="0" />
      <contractDBID value="111" />
    </CfgObjectiveTableRecord>
  </objectiveRecords>
  <state value="1" />
  <userProperties>
    <list_pair key="_WIZARD_">
      <str_pair key="_ProgID_" value="CfgWizard_CBR.CfgObjectiveTableWizard" />
    </list_pair>
  </userProperties>
  <prepaidCost value="725" />
  <timeZoneDBID value="115" />
  <timeStart value="1169649000" />
  <timeEnd value="1172447999" />
  <type value="2" />
</CfgObjectiveTable>
```

See Also

[CfgDeltaObjectiveTable](#)

[CfgEnumeratorValue](#)

CfgObjectiveTableRecord

Description

Objective Records define records within Objective Table objects. The `mediaTypeDBID`, `serviceTypeDBID` and `customerSegmentDBID` properties are mandatory. They compose a unique key for these records within the Objective Table that contains them.

Attributes

- `mediaTypeDBID` — A unique identifier of the **Media Type** to which this objective table record is allocated. Mandatory. Once specified, cannot be changed. Only enumerator values belonging to the enumerator with name `Media Type` are allowed in this field.
- `serviceTypeDBID` — A unique identifier of the **Service Type** this objective table record is associated with. Mandatory. Once specified, cannot be changed. Only enumerator values belonging to the enumerator with name `Service Type` are allowed in this field.
- `customerSegmentDBID` — A unique identifier of the **Customer Segment** this objective table record is associated with. Mandatory. Once specified, cannot be changed. Only enumerator values belonging to the enumerator with name `Customer Segment` are allowed in this field.
- `objectiveTreshold` — An objective threshold value for this record.
- `objectiveDelta` — An objective delta value for this record. Defines the step of objective threshold deviation
- `contractDBID` — An **IT Contract** associated with this Objective Record.

Comments

The following table presents the default `objectiveThreshold` values for corresponding media types.

| Media Type | objectiveThreshold (default values) |
|------------|-------------------------------------|
| Voice | 20 sec |
| voip | 20 sec |
| Email | 24 hours (60x60x24=86400 sec) |
| Vmail | 24 hours (60x60x24=86400 sec) |
| Smail | 24 hours (60x60x24=86400 sec) |
| Chat | 20 sec |
| Video | 20 sec |
| Cobrowsing | - |

| MediaType | objectiveThreshold (default values) |
|------------------|--|
| Whiteboard | - |
| Appsharing | - |
| Webform | 24 hours (60x60x24=86400 sec) |
| Workitem | 24 hours (60x60x24=86400 sec) |
| Callback | 1 hour (3600 sec) |
| Fax | 24 hours (60x60x24=86400 sec) |
| Imchat | 20 sec |
| Busevent | 1 hour (60x60=3600 sec) |
| Alert | 20 sec |
| Sms | 20 sec |
| Any | 24 hours (60x60x24=86400 sec) |

CfgPerson

Description

Persons correspond to contact center personnel — including agents — who need access to CTI applications.

The Genesys Framework requires that every person who needs such access be registered in the Configuration Database with an appropriate set of privileges.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|---|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the persons that belong to this tenant. |
| is_agent | int | Indicator of whether a person is an agent. If set to <code>CFGTrue</code> , Configuration Server will return information only about the persons who are agents. If set to <code>CFGFalse</code> , Configuration Server will return information only about the persons who are not agents. |
| skill_dbid | int | A unique identifier of a skill. If specified, Configuration Server will return information only about the agents who have this skill. |
| group_dbid | int | A unique identifier of an agent group. If specified, Configuration Server will return information only about the agents who form this group. |
| state | int | Current state of a person (see type <code>CfgObjectState</code>). If specified, Configuration Server will return information only about persons that are currently in this state. |
| employee_id | string | Employee ID of a person. Shall be specified as a character string. If specified, Configuration Server will return information only about the person(s) with this employee |

| Filter Name | Type | Description |
|---------------|--------|--|
| | | ID. |
| login_dbid | int | A unique identifier of an agent login. If specified, Configuration Server will return information only about the agent this login is currently assigned to. |
| user_name | string | User name of a person. Shall be specified as a character string. If specified, Configuration Server will return information only about the person with this user name. |
| dbid | int | A unique identifier of a person. If specified, Configuration Server will return information only about this person. |
| no_login_dbid | int | Configuration Server will return information only about the agent(s) without login is currently assigned to. |
| no_place_dbid | int | Configuration Server will return the information only about the agents that do not have default places associated with. |
| first_name | string | The name of a person. Shall be specified as a character string. If specified, Configuration Server will return information only about the person with this name. |
| last_name | string | The last name of a person. Shall be specified as a character string. If specified, Configuration Server will return information only about the person with this last name. |
| switch_dbid | int | A unique identifier of a Switch. If specified, Configuration Server will return information only about the agent(s) that have associated Agent Logins belonged to that Switch. |

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 **Tenant** whose employee this person is. Once specified, cannot be changed.

- `lastName` — A pointer to the person's last name. Max length 64 symbols.
- `firstName` — A pointer to the person's first name. Max length 64 symbols.
- `address` — Not in use.
- `phones` — Not in use.
- `birthdate` — Not in use.
- `comment` — Not in use.
- `employeeID` — A pointer to the code identifying this person within the tenant staff. Mandatory. Must be unique within the tenant. Max length 64 symbols.
- `userName` — A pointer to the name the person uses to log into a CTI system. Mandatory. Must be unique within the Configuration Database.
- `password` — A pointer to the password the person uses to log into a CTI system. Max length 64 symbols.
- `appRanks` — A pointer to the list of the person's ranks with respect to applications (every item of this list is structured as `CfgAppRank`). When used as an entry in `CfgDeltaPerson` (see below), it is a pointer to a list of the ranks added to the existing list.
- `isAgent` — An indicator of whether the person is an agent. Read-only (set automatically according to the current value of `agentInfo` below). See type `CfgFlag`.
- `agentInfo` — A pointer to the structure containing agent-specific information. See structure `CfgAgentInfo`. Shall be specified if the person is an agent and shall be set to NULL otherwise. Once specified, cannot be set to NULL.
- `isAdmin` — Not in use.
- `assignedTenantDBIDs` — Not in use.
- `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.
- `emailAddress` — A pointer to the email address of this person. Max length 255 symbols.
- `externalID` — A pointer to the string used to identify this person in the external systems. In particular, this field used to store an identifier processed during the authentication in the LDAP repositories. Max length 255 symbols.

Comments

Whether a new person is an agent or not shall be specified at the time when the corresponding `CfgPerson` object is created. It is not possible to change person's status from a non-agent to an agent (or the other way around) once the `CfgPerson` object has been created.

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

- Modification of `agentDBIDs` of all agent groups that included Person X as an agent
- Modification of `memberIDs` of all access groups that included Person X as a member

- Deletion of Person X

A person cannot be deleted as long as it is associated as an account with at least one daemon application (See [CfgApplication](#) and [ConfSetAccount](#)).

By default, access privileges of a new person will be set according to the following rules:

- Any non-agent of the Service Provider becomes a member of the default access group Administrators of the Service Provider.
- Any agent of the Service Provider becomes a member of the default access group Users of the Service Provider.
- Any non-agent of a particular tenant becomes a member of the default access group Administrators of that tenant.
- Any agent of a particular tenant becomes a member of the default access group Users of that tenant.
- Any person added to the Configuration Database is also considered a member of the access group Everyone, which cannot be changed. For specification of access privileges of the above default groups, refer to comments to object CfgAccessGroup in section Access Control Functions and Data Types.
- Person with DBID = 100 and tenantDBID = 1 shall be pre-defined (scripted) in the Configuration Database before Configuration Server is started for the first time. This person has Full Control permissions with respect to all objects in the Configuration Database, which cannot be changed. The object that represents this person cannot be deleted.

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgPerson>
  <DBID value="125" />
  <tenantDBID value="101" />
  <lastName value="Name" />
  <firstName value="My" />
  <employeeID value="001" />
  <userName value="001" />
  <password value="FFFFFFFF" />
  <isAgent value="2" />
  <CfgAgentInfo>
    <placeDBID value="112" />
    <agentLogins>
      <CfgAgentLoginInfo>
        <agentLoginDBID value="147" />
        <wrapupTime value="0" />
      </CfgAgentLoginInfo>
    </agentLogins>
    <capacityRuleDBID value="0" />
    <siteDBID value="0" />
    <contractDBID value="0" />
  </CfgAgentInfo>
</CfgPerson>
```

```
<isAdmin value="1" />
<state value="1" />
<emailAddress value="My.Name@my.host.com" />
</CfgPerson>
```

See Also

[CfgDeltaPerson](#)

[CfgSkill](#)

[CfgAgentLogin](#)

[CfgAgentGroup](#)

CfgPhones

Attributes

- office — A pointer to the office phone number. Max length 64 symbols.
- home — A pointer to the home phone number. Max length 64 symbols.
- mobile — A pointer to the mobile phone number. Max length 64 symbols.
- pager — A pointer to the pager number. Max length 64 symbols.
- fax — A pointer to the fax number. Max length 64 symbols.
- modem — A pointer to the modem number. Max length 64 symbols.
- phonesComment — A pointer to the comment line. Max length 64 symbols.

CfgPhysicalSwitch

Description

Switching Offices are the actual telephone switches that provide telephone service to contact centers.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|---|
| dbid | int | A unique identifier of a physical switch. If specified, Configuration Server will return information only about this physical switch. |
| name | string | Name of a physical switch. Shall be specified as a character string. If specified, Configuration Server will return information only about the physical switch with that name. |
| state | int | Current state of a physical switch (see CfgObjectState). If specified, Configuration Server will return information only about physical switches that are currently in this state. |
| folder_dbid | int | A unique identifier of a folder. If specified, Configuration Server will return information only about the physical switches located immediately under this folder. |

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.
- **name** — A pointer to the name of the switch. Mandatory. Must be unique within the Configuration Database.
- **type** — Type of this physical switch. Mandatory. Once specified, cannot be changed. See [CfgSwitchType](#).
- **address** — Not in use.

- `contactPersonDBID` — Not in use.
- `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

A physical switch cannot be deleted as long as it is associated with at least one switch (see [CfgSwitch](#)).

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgPhysicalSwitch>
  <DBID value="101" />
  <name value="ASwitch" />
  <type value="12" />
  <contactPersonDBID value="0" />
  <state value="1" />
</CfgPhysicalSwitch>
```

See Also

[CfgDeltaPhysicalSwitch](#)

CfgPlace

Description

A *Place* is a location that has one or more DNs operated by a single agent.

Configure Places and assign individual DNs to them in order to monitor performance and availability of Agents, Agent Groups, and Place Groups, and to provide this information to call-processing applications.

A typical Agent Place consists of two DNs: one DN that an agent uses to take customer calls and another DN the agent uses to make consultation calls and transfers. If you are using the multimedia options of the Genesys products, Places may have to be equipped with DNs of other types, such as E-mail Address.

Make sure the configuration of Places in the Configuration Database always matches the actual wiring arrangements in the contact center.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|--|
| dbid | int | A unique identifier of a place. If specified, Configuration Server will return information only about this place. |
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the places that belong to this tenant. |
| name | string | Name of a place. Shall be specified as a character string. If specified, Configuration Server will return information only about the place(s) with that name. |
| state | int | Current state of a place (see CfgObjectState). If specified, Configuration Server will return information only about places that are currently in this state. |
| group_dbid | int | A unique identifier of a place group. If specified, Configuration Server will return information only about the places that form this group. |

| Filter Name | Type | Description |
|-------------|------|---|
| dn_dbid | int | A unique identifier of a DN. If specified, Configuration Server will return information only about the place this DN is assigned to. |
| no_dn_dbid | int | Configuration Server will return information only about the place(s) without DNs assigned to. |
| person_dbid | int | A unique identifier of a person. If specified, Configuration Server will return information only about place associated with this person. |

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 **Tenant** that this place belongs to. Mandatory. Once specified, cannot be changed.
- **name** — A pointer to the name of the place. Mandatory. Must be unique within the tenant.
- **DNDBIDs** — A pointer to the list of identifiers of the telephony objects that are assigned to this place. When used as an entry in **CfgDeltaPlace** (see below), it is a pointer to a list of identifiers of the objects added to the existing list. DNs assigned to the place must belong to the tenant specified by **tenantDBID** above. One DN cannot be assigned to more than one place. DNs of the following types cannot be included into a place: **CFGACDQueue**, **CFGRoutingPoint**, **CFGVirtACDQueue**, **CFGVirtRoutingPoint**, **CFGTrunk**, **CFGTrunkGroup**, **CGFTieLine**, **CGFTieLineGroup**, **CFGExtRoutingPoint** and **CFGRoutingQueue**. **CFGDestinationLabel**, **CFGServiceNumber**, **CFGAccessResource**.
- **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.
- **capacityRuleDBID** — A unique identifier of the capacity rule (**CfgScript**) associated with this place.
- **siteDBID** — A unique identifier of Site (**CfgFolder**) with which this Place is associated.
- **contractDBID** — A unique identifier of Cost Contract (**CfgObjectiveTable**) with which this Place is associated.

Comments

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

- Modification of pplaceDBIDs of all place groups that included Place X
- Modification of pplaceDBID of all persons who had Place X assigned as the default place
- Deletion of Place X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgPlace>
  <DBID value="101" />
  <tenantDBID value="1" />
  <name value="001" />
  <DNDBIDs>
    <DBID value="102" />
  </DNDBIDs>
  <state value="1" />
  <capacityRuleDBID value="0" />
  <siteDBID value="0" />
  <contractDBID value="0" />
</CfgPlace>
```

See Also

[CfgDeltaPlace](#)

[CfgDN](#)

[CfgPlaceGroup](#)

CfgPlaceGroup

Description

You can group Places if, according to the call-processing algorithms, the calls have to be distributed among a set of Places under the control of CTI applications rather than through the ACD mechanisms of the Switch.

As an example, consider a call-parking service, where a routing application transfers a call to one of the ports assigned to a call-parking Place Group and attaches the information about the treatment to be applied to that call while it is parked.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|--|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the place groups that belong to this tenant. |
| place_dbid | int | A unique identifier of a place. If specified, Configuration Server will return information only about the place groups that include this place. |
| state | int | Current state of a place group (see CfgObjectState). If specified, Configuration Server will return information only about place groups that are currently in this state. |
| name | string | Name of a place group. Shall be specified as a character string. If specified, Configuration Server will return information only about the place group(s) with that name. |
| dbid | int | A unique identifier of a group. If specified, Configuration Server will return information only about this group. |

Attributes

- groupInfo — A pointer to the structure [CfgGroup](#) containing general information about this group.

Mandatory.

- placeDBIDs — A pointer to the list of identifiers of the **Places** that form this group. Note: Configuration Server does not place any restrictions regarding the types of DNs in the places that form a group.

Comments

The name of a place group must be unique within the tenant, but can coincide with the name of either an agent group or a DN group of the same tenant.

The name of a place group cannot be changed until there is at least one place listed in this group. See placeDBIDs property

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

- Modification of campaignGroups of all campaigns that included Place Group X
- Deletion of Place Group X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgPlaceGroup>
  <CfgGroup>
    <DBID value="106" />
    <tenantDBID value="101" />
    <name value="APlaceGroup" />
    <capacityTableDBID value="0" />
    <quotaTableDBID value="0" />
    <state value="1" />
    <capacityRuleDBID value="0" />
    <siteDBID value="0" />
    <contractDBID value="0" />
  </CfgGroup>
  <placeDBIDs>
    <DBID value="111" />
    <DBID value="112" />
  </placeDBIDs>
</CfgPlaceGroup>
```

See Also

[CfgDeltaPlaceGroup](#)

CfgPlace

CfgPortInfo

Description

CfgPortInfo contains information about a listening port for a server.

Attributes

- id — An identifier of the server's listening port.
- port — Listening port value.
- transportParams — Listening port's transport parameters
- connProtocol — A pointer to the name of the connection control protocol.
- appParams — Listening port's application parameters
- description — Optional description of the listening port
- charField1 — Optional text field #1
- charField2 — Optional text field #2
- charField3 — Optional text field #3
- charField4 — Optional text field #4
- longField1 — Optional integer field #1
- longField2 — Optional integer field #2
- longField3 — Optional integer field #3
- longField4 — Optional integer field #4

CfgRole

Description

A *Role* defines what tasks, if any, a user (either an individual User, or member of an Access Group) can or cannot perform on an object for which that user has corresponding permissions.

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 [Tenant](#) that this role belongs to. Mandatory. Once specified, cannot be changed.
- **name** — The name of the role. Mandatory. Must be unique within the tenant.
- **description** — A text description of this role.
- **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.
- **members** — A list of members belonging to this role. Refer to [CfgRoleMember](#).

See Also

[CfgRoleMember](#)

[CfgDeltaRole](#)

CfgRoleMember

Description

A *RoleMember* is a structure used to describe a Person or Access Group that belongs to a particular role.

Attributes

- `objectDBID` — A unique identifier of the [Person](#) or [Access Group](#) for this role member.
- `objectType` — The [Object Type](#) of this role member.

See Also

[CfgRole](#)

[CfgDeltaRole](#)

CfgScheduledTask

Description

A *Scheduled Task* is a GVP Object in which you schedule the GVP tasks

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|--|
| task_type | int | Type of the ScheduledTask (see CfgTaskType). If specified, Configuration Server will return information only about the ScheduledTasks of this type. |
| state | int | Current state of a ScheduledTask (see CfgObjectState). If specified, Configuration Server will return information only about ScheduledTasks that are currently in this state. |
| name | string | Name of a ScheduledTask. Shall be specified as a character string. If specified, Configuration Server will return information only about the ScheduledTask(s) with that name. |
| dbid | int | A unique identifier of a ScheduledTask. If specified, Configuration Server will return information only about this ScheduledTask. |

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.
- **name** — A pointer to name of the *ScheduledTask*. Mandatory. Must be unique within the Configuration Database.
- **type** — Type of the *ScheduledTask*. Mandatory. Once specified, cannot be changed. See type [CfgTaskType](#).

- `description` — An optional short description of this *ScheduledTask*.
- `startTime` — The time the task is to begin.
- `resources` — A pointer to the list of the objects associated with this *ScheduledTask* (every item of this list is structured as [CfgObjectResource](#)).
When used as an entry in [CfgDeltaScheduledTask](#), it is a pointer to a list of resources added to the existing list. Only objects of the following types can be associated with *ScheduledTask* object through resources:
 - [CfgHost](#)
 - [CfgSwitch](#)
 - [CfgGVPCustomer](#)
- `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.

See Also

[CfgDeltaScheduledTask](#)

CfgScript

Description

Scripts identify processing scenarios or treatments that can be applied to customer interactions.

Filter Keys

| Filter Name | Type | Description |
|---------------------------|--------|--|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the script(s) that belong to this tenant. |
| script_type | int | Type of the script (see CfgScriptType). If specified, Configuration Server will return information only about the script(s) of this type. |
| state | int | Current state of a script (see CfgObjectState). If specified, Configuration Server will return information only about scripts that are currently in this state. |
| dbid | int | A unique identifier of a script. If specified, Configuration Server will return information only about this script. |
| name | string | Name of a script. Shall be specified as a character string. If specified, Configuration Server will return information only about the script(s) with that name. |
| capacity_tenant_dbid | int | A unique identifier of Tenant. If specified, Configuration Server will return information only about the script that is associated with specified tenant. |
| capacity_agent_dbid | int | A unique identifier of Person/ Agent. If specified, Configuration Server will return information only about the script that is associated with specified agent. |
| capacity_agent_group_dbid | int | Unique identifier of an |

| Filter Name | Type | Description |
|---------------------------|------|---|
| | | AgentGroup. If specified, Configuration Server will return information only about the script that associated with agent group. |
| capacity_place_dbid | int | Unique identifier of a Place. If specified, Configuration Server will return information only about the script that is associated with specified place. |
| capacity_place_group_dbid | int | Unique identifier of a PlaceGroup. If specified, Configuration Server will return information only about the script that is associated with specified place group. |
| exclude_bytecode | int | A flag controlling how the bytecode binary option from the userPropertiesInsert text field will be returned. If set in the filter, Configuration Server will return an empty list under this option without regard to the actual content. |

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.
- **name** — A pointer to the name of the script. Mandatory. Must be unique within the tenant.
- **tenantDBID** — A unique identifier of the [CfgTenant](#) Tenant this script belongs to. Mandatory. Once specified, cannot be changed.
- **index** — Script index.
- **type** — Type of this script. Mandatory. Once specified, cannot be changed. See type [CfgScriptType](#).
- **contactPersonDBID** — Not in use.
- **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.
- **resources** — A pointer to the list of the objects associated with this Host (every item of this list is structured as [CfgObjectResource](#)).
When used as an entry in [CfgDeltaHost](#), it is a pointer to a list of resources added to the existing list. Only objects of the following types can be associated with Script object through resources:
 - [CfgAppPrototype](#)
 - [CfgScript](#)

Comments

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

- Modification of campaigns that had scriptDBID field set to Script X
- Modification of campaign groups that had scriptDBID field set to Script X
- Modification of calling lists that had scriptDBID field set to Script X
- Modification of voice prompts that had scriptDBID field set to Script X
- Modification of alarm conditions that had scriptDBID field set to Script X
- Modification of alarm conditions that had to Script X included in their reactionScriptDBIDs or clearanceScriptDBIDs field
- Modification of agents that had capacityRuleDBID field set to Script X
- Modification of places that had capacityRuleDBID field set to Script X
- Modification of agent groups that had capacityRuleDBID field set to Script X
- Modification of place groups that had capacityRuleDBID field set to Script X
- Modification of tenants that had defaultCapacityRuleDBID field set to Script X
- Deletion of Script X

XML Representation

Important

This XML was created using the Configuration Server 7.5 schema.2

```
<CfgScript>
  <DBID value="101" />
  <name value="Script" />
  <tenantDBID value="101" />
  <index value="0" />
  <type value="5" />
  <contactPersonDBID value="0" />
  <state value="1" />
  <userProperties>
    <list_pair key="Property">
      <str_pair key="key" value="value" />
    </list_pair>
  </userProperties>
</CfgScript>
```

See Also

[CfgDeltaScript](#)

CfgServer

Description

A server.

Attributes

- `hostDBID` — A unique identifier of the [CfgHost](#) where this server resides. Cannot be changed as long as the server is associated with at least one client application or a primary server.
- `port` — A pointer to the name of the port which client applications should use to open communication sessions to this server. Populated for backward compatibility purpose from `portInfos` list.
- `backupServerDBID` — An identifier of the [Server](#) which is to be contacted if connection to this server fails. The backup server must be associated with the same account (see `ConfSetAccount`) and have the same application type (`CfgAppType`). One backup server cannot be associated with more than one primary server. See comments.
- `timeout` — Time-out in seconds that the application should run before making a re-connect attempt after a communication session with this server has failed. May not be set to a negative value. Recommended to be set to 10 by default.
- `attempts` — Number of attempts to connect to this server before trying to connect to the backup server. Makes sense only if `backupServerDBID` is specified. May not be set to a negative value. Recommended to be set to 1 by default.

Comments

When an application is designated as a backup server for another server, values of the following parameters of this application will be automatically changed to match the values of the same parameters of the primary server:

- `appServerDBIDs`
- `tenantDBIDs`
- `flexibleProperties`

As long as this application is associated with the primary server, these parameters will be treated as read-only, and their values will be changed only when changes are applied to the corresponding parameters of the primary server.

CfgService

Description

Solutions are sets of functions that applications provide. Solutions accomplish particular business tasks in contact centers.

Filter Keys

| Filter Name | Type | Description |
|-------------|------|--|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the services this tenant is subscribed to. |
| app_type | int | Type of the application (see CfgAppType). If specified, Configuration Server will return information only about the services that involve this application type. |
| state | int | Current state of a service (see CfgObjectState). If specified, Configuration Server will return information only about services that are currently in this state. |
| app_dbid | int | A unique identifier of an application. If specified, Configuration Server will return information only about the Solutions that involve this application. |
| scs_dbid | int | A unique identifier of a Service Control Server. If specified, Configuration Server will return information only about the solutions controlled by this SCS. |
| type | int | The type of the solution. If specified, Configuration Server will return information only about the solutions of specified type. |
| folder_dbid | int | A unique identifier of a folder. If specified, Configuration Server will return information only about |

| Filter Name | Type | Description |
|-------------|------|---|
| | | the services located immediately under this folder. |

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.
- **name** — A pointer to the name of the service/solution. Mandatory. Must be unique within the Configuration Database.
- **abbr** — Not in use.
- **type** — Not in use.
- **appServicePermissions** — Not in use.
- **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.
- **solutionType** — The type of the solution. Mandatory. Once specified cannot be changed. See [CfgSolutionType](#).
- **components** — A pointer to a list of solution components defined for this solution (every item of this list is structured as [CfgSolutionComponent](#)).
When used as an entry in [CfgDeltaService](#), it is a pointer to a list of solution components added to the existing list.
- **SCSDBID** — A unique identifier of an [Application](#) of CFGSCStype which is supposed to control the solution.
See comments. See also [CfgSolutionComponent](#).
- **assignedTenantDBID** — A unique identifier of the [Tenant](#) the solution is assigned to. Optional.
If `assignedTenantDBID` is managed (added/modified/removed) by configuration wizard, the same action (add/modify/remove) has to be initiated by wizard for property `tenantDBIDs` of [CfgApplication](#) if an application belongs to this solution.
- **version** — A pointer to the version of the solution. Mandatory.
- **componentDefinitions** — A pointer to a list of predefined solution components for this solution. (Every item of this list is structured as a [CfgSolutionComponentDefinition](#).)
- **startupType** — A type of solution/service startup. Specifies whether this solution/service have to be started by Management Layer. See [CfgStartupType](#). Read-only. Specified during application prototype definition.
The value is associated with solution type [CfgSolutionType](#). The value for the solutions of CFGSTDefaultSolutionType and CFGSTFramework type is set to CFGSUTManual, and for other applications is set to CFGSUTAutomatic.

Comments

The components list of a solution can be populated based on the list of the componentDefinitions the solution is based on. Then, an application for each component within the solution should be assigned according to the type of application specified in SolutionComponentDefinition this solution component is based on and whether or not a solution component is mandatory or optional. Parameters appType, appVersion, and startupPriority are READONLY within solution.

It shall be possible to edit the components list of a solution in such a way that one or more copies of a solution component already defined within the solution can be created. If such a copy is made, a unique identifier of an application with the type suitable for the newly created component could be assigned to appDBID parameter in this component.

The fact that more than one solution component within a solution may have the same appType, appVersion and startupPriority implies that the order of activation of corresponding applications within the solution may be chosen arbitrarily.

For compatibility purposes between 5.1.xx and 5.9.xxx, objects of type [CfgService](#) that exist in 5.1.xxx release will have solutionType=CFGSTDefaultSolutionType.

If one solution component is used by two different solutions, both solutions should refer to same application of CFGSCS type (SCSDBID)

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgService>
  <DBID value="114" />
  <name value="Service1" />
  <type value="2" />
  <state value="1" />
  <solutionType value="15" />
  <SCSDBID value="134" />
  <assignedTenantDBID value="1" />
  <version value="7.5" />
  <startupType value="1" />
</CfgService>
```

See Also

[CfgDeltaService](#)

[CfgTenant](#)

CfgApplication

CfgServiceInfo

Attributes

- `serviceDBID` — A unique identifier of the service. See [CfgService](#). Mandatory. Once specified, cannot be changed. The same value cannot be repeated within one list.
- `isChargeable` — An indicator of whether the tenant is or is not to be charged for this service (see [CfgFlag](#)). Recommended to be set to `CFGTrue` by default.

CfgSkill

Description

Skills are qualities or abilities that Agents possess. These Skills affect how Agents are placed in a contact center hierarchy.

Common skills include abilities in different languages, particular categories of product knowledge, or ability in particular types of sales.

[Top of Page](#)

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|--|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the skill(s) that belong to this tenant. |
| state | int | Current state of a skill (see CfgObjectState). If specified, Configuration Server will return information only about skills that are currently in this state. |
| dbid | int | A unique identifier of a skill. If specified, Configuration Server will return information only about this skill. |
| name | string | Name of a skill. Shall be specified as a character string. If specified, Configuration Server will return information only about the skill(s) with that name. |

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.
- name — A pointer to the name of the skill. Mandatory. Must be unique within the tenant.

- tenantDBID — A unique identifier of the **Tenant** that this skill belongs to. Mandatory. Once specified, cannot be changed.
- 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

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

Modification of skillLevels of all agents that were associated with Skill X
Deletion of Skill X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgSkill>
  <DBID value="101" />
  <name value="Skill" />
  <tenantDBID value="1" />
  <state value="1" />
</CfgSkill>
```

See Also

CfgDeltaSkill

CfgSkillLevel

Description

CfgSkillLevel indicates an Agent's level of proficiency in a particular skill.

Attributes

- **skillDBID** — A unique identifier of the skill the level relates to. See [CfgSkill](#). Mandatory. Once specified, cannot be changed. This skill must belong to the tenant that this agent belongs to. The same value cannot be repeated within one list.
- **level** — Level of the skill. Cannot be a negative value.

CfgSolutionComponent

Description

CfgSolutionComponent displays a list of applications whose functionality this solution uses.

Attributes

- `startupPriority` — The startup priority of the solution component in a component sequence. Component numbers should be used to determine the order in which components should be started and stopped. Must be defined as positive integer. Mandatory. Once specified cannot be changed. Must be shown as READONLY property in object of type `CfgService` (Solution).
- `isOptional` — Determines whether this solution component is optional. Recommended to be set to `CFGFalse` by default. Must be shown as READONLY property in object of type `CfgService` (Solution). Refers to `CfgFlag`.
- `appDBID` — A unique identifier of an `Application` with type `appType` and version `appVersion`. Mandatory (application must be specified within solution) for solution components within a solution if `isOptional` is set to `CFGFalse`. The application could be chosen from the list of applications based on application template/prototype specified in `appPrototypeDBID`.

The following rule should be used during the creation of a solution: an application can participate in different solutions.

CfgSolutionComponentDefinition

Description

CfgSolutionComponentDefinition lists the types of applications whose functionality this solution uses.

Attributes

- **startupPriority** — The default number of the solution component in a component startup sequence within solution. **StartupPriority** value should be used to determine the order in which components should be started and stopped. The value defined as default could be changed at time of **SolutionComponent** definition. Must be defined as positive integer. Mandatory. Must be shown as the **READONLY** property in an object of type **CfgService** (Solution).
- **isOptional** — Determines whether this solution component default value is optional. Recommended to be set to **CFGFalse** by default. Must be shown as **READONLY** property in object of type **CfgService** (Solution).
Refers to **CfgFlag**.
- **type** — Type of the application that is used as solution component. Mandatory. Once specified, cannot be changed. See **CfgAppType**.
- **version** — A pointer to the application version that is used as solution component. Once specified, cannot be changed.

CfgStatDay

Description

A *Statistical Day* is a numerically-expressed workload that a particular Agent Group is expected to handle during a particular business day.

Filter Keys

| Filter Name | Type | Description |
|--------------|--------|--|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the days that are defined within this tenant. |
| table_dbid | int | A unique identifier of a stat table. If specified, Configuration Server will return information only about the days that are defined within this table. |
| state | int | Current state of a stat day (see CfgObjectState). If specified, Configuration Server will return information only about stat days that are currently in this state. |
| dbid | int | A unique identifier of a day. If specified, Configuration Server will return information only about this day. |
| name | string | Name of a day. Shall be specified as a character string. If specified, Configuration Server will return information only about the day(s) with that name. |
| statday_type | int | Type of a stat day (see CfgStatDayType). If specified, Configuration Server will return information only about stat days have this type. |

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 **Tenant** that this day is defined for. Mandatory. Once specified, cannot be changed.
- **name** — A pointer to the name of the day. Mandatory. Must be unique within the tenant.
- **isDayOfWeek** — Indicator of whether this day is identified as a day of week (CFGTrue) or a day of year (CFGFalse). Mandatory. Once specified, cannot be changed. The parameter is ignored if value of parameter **date** is specified. Refer to **CfgFlag**.
- **day** — Day's number. If **isDayOfWeek** is set to CFGTrue, the allowable range is from 1 to 7 (where 1 stands for Sunday). If **isDayOfWeek** is set to CFGFalse, the allowable range is from 1 to 366 and -1 (where 1 stands for January 1 and -1 stands for any day). Mandatory. Once specified, cannot be changed. The parameter is ignored if value of parameter **date** is specified.
- **startTime** — Start of business time of day measured in minutes from 00:00. Cannot be negative and cannot be greater than 1440.
- **endTime** — End of business time of day measured in minutes from 00:00. Cannot be negative and cannot be greater than 1440. If set to be less than the setting for **startTime**, implies the time of the next calendar day (night shift).
- **minValue** — Minimum statistical value for the day. Cannot be negative.
- **maxValue** — Maximum statistical value for the day. Cannot be negative or less than the setting for **minValue**.
- **targetValue** — Target statistical value for the day. Cannot be less than the setting for **minValue** or greater than the setting for **maxValue**.
- **intervalLength** — Statistical interval in minutes. Must be a multiple of five. Once specified, cannot be changed. Recommended to be set to 15 by default.
- **statIntervals** — A pointer to the list of the structures of **CfgStatInterval** type that associate intervals with certain statistical values.
When used as an entry in **CfgDeltaStatDay**, it is a pointer to a list of structures of type **CfgStatInterval** added to the existing list.
- **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.
- **date** — Day of specific year calculated since 00:00:00 GMT Jan 1, 1970 measured in seconds. Once specified, cannot be changed. If value is specified the values of parameters **isDayOfWeek** and **day** must be ignored.
- **type** — Statistical Day type. Refer to **CfgStatDayType**.
- **useFlatRate** — Flag determining whether **flatRate** (CFGTrue) should be selected. Refer to **CfgFlag**.
- **flatRate** — Amount to be charged for processing the forecast volume of interactions.

Comments

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

- Modification of statDayDBIDs of all stat tables that included Stat Day X
- Deletion of Stat Day X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgStatDay>
  <DBID value="105" />
  <tenantDBID value="101" />
  <name value="az" />
  <isDayOfWeek value="1" />
  <day value="303" />
  <startTime value="480" />
  <endTime value="1020" />
  <minValue value="0" />
  <maxValue value="100" />
  <targetValue value="50" />
  <intervalLength value="15" />
  <statIntervals>
    <CfgStatInterval>
      <intervalCount value="1" />
      <statValue1 value="22" />
      <statValue2 value="2" />
      <statValue3 value="2" />
      <statValue4 value="0" />
    </CfgStatInterval>
  </statIntervals>
  <state value="1" />
  <userProperties>
    <list_pair key="Property">
      <str_pair key="key" value="value" />
    </list_pair>
  </userProperties>
  <date value="0" />
  <type value="2" />
  <useFlatRate value="1" />
  <flatRate value="0" />
</CfgStatDay>
```

See Also

[CfgDeltaStatDay](#)

[CfgStatTable](#)

CfgStatInterval

Description

Use a *Statistical Interval* to associate each Statistical Interval with certain Statistical Values.

Attributes

- `intervalCount` - Interval's number. Once specified, cannot be changed. The first interval has number 1. First interval is always counted from the start of business time set by `startTime` in [CfgStatDay](#).
- `statValue1` - Statistical values for this interval. Cannot be negative.
- `statValue2` - Statistical values for this interval. Cannot be negative.
- `statValue3` - Statistical values for this interval. Cannot be negative.
- `statValue4` - Statistical values for this interval. Cannot be negative.

Comments

For the missing intervals, the stat values shall be assumed to be zero.

The current version of Configuration Server does not verify correspondence between the values of `intervalCount` and the current settings of `intervalLength`, `startTime` and `endTime` in [CfgStatDay](#). Such verification may be added in one of the next releases. At the moment, it is users' responsibility to make sure the values of `intervalCount` make sense.

CfgStatTable

Description

Statistical Tables are groups of Statistical Days that represent statistically-modeled performances of Agent Groups over a calendar period of up to one year.

Call-processing applications can use Statistical Tables to provide load balancing between Agent Groups when the real-time statistics for those groups are unavailable.

Filter Keys

| Filter Name | Type | Description |
|---------------|--------|---|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the tables that are defined within this tenant. |
| table_type | int | Type of the table. If specified, Configuration Server will return information only about the tables of this type. |
| group_dbid | int | A unique identifier of a group. If specified, Configuration Server will return information only about the tables assigned to this group. |
| state | int | Current state of a stat table (see CfgObjectState). If specified, Configuration Server will return information only about tables that are currently in this state. |
| dbid | int | A unique identifier of a table. If specified, Configuration Server will return information only about this table. |
| name | string | Name of a table. Shall be specified as a character string. If specified, Configuration Server will return information only about the table(s) with that name. |
| stat_day_dbid | int | A unique identifier of a stat day. If specified, Configuration Server will return information only about the tables containing this stat |

| Filter Name | Type | Description |
|-------------|------|-------------|
| | | day. |

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 **Tenant** that this table is defined for. Mandatory. Once specified, cannot be changed.
- **name** — A pointer to the name of the table. Mandatory. Must be unique within the tenant.
- **type** — Type of stat table. Once specified, cannot be changed. Refer to **CfgStatTableType**.
- **statDayDBIDs** — A pointer to the list of identifiers of the **Statistical Days** that constitute the table. When used as an entry in **CfgDeltaStatTable** (see below), it is a pointer to a list of days added to the existing list.
- **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.
- **waitThreshold** — The time the Universal Routing Server can wait until the corresponding resources become available.
- **flatRate** — A flat rate measured in cost units.
- **agentHourlyRate** — The hourly rate for an agent. This value is used only for Variable Rate Contracts.
- **useFlatRate** — Flag determining whether `flatRate` (CFGTrue) or `agentHourlyRate` (CFGFalse) should be selected. Refer to **CfgFlag**.

Comments

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

- Modification of `capacityTableDBID` or `quotaTableDBID` of all agent groups that were associated with Stat Table X
- Modification of `capacityTableDBID` or `quotaTableDBID` of all place groups that were associated with Stat Table X
- Modification of `capacityTableDBID` or `quotaTableDBID` of all DN groups that were associated with Stat Table X
- Deletion of Stat Table X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgStatTable>
  <DBID value="106" />
  <tenantDBID value="101" />
  <name value="az" />
  <type value="5" />
  <state value="1" />
  <userProperties>
    <list_pair key="Property">
      <str_pair key="key" value="value" />
    </list_pair>
  </userProperties>
  <waitThreshold value="0" />
  <flatRate value="0" />
  <agentHourlyRate value="0" />
  <useFlatRate value="2" />
</CfgStatTable>
```

See Also

[CfgDeltaStatTable](#)

[CfgStatDay](#)

CfgSubcode

Description

You can use *Subcodes* to supplement Action Codes with more precise information about the reasons for a certain action.

Attributes

- **name** — A pointer to the name of the subcode. Mandatory. Must be unique within the action code. Once specified, cannot be changed.
- **code** — A pointer to the index or abbreviation of the subcode. Mandatory. Once specified, cannot be changed.

CfgSwitch

Description

A *Switch* is an aggregate of telephony resources within a Switching Office.

Most enterprise-level configurations have a one-to-one match between switches and switching offices. However, there may be instances when it is desirable to partition an office into more than one switch, perhaps due to CTI-link capacity limitations, or to create a more efficient and secure numbering plan. In that case, you must define these switches within a switching office.

Filter Keys

| Filter Name | Type | Description |
|--------------|------|--|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the switches that belong to this tenant. |
| tserver_dbid | int | A unique identifier of a T-Server. If specified, Configuration Server will return information only about the switch that is associated with this T-Server. |

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 Tenant to which this switch is allocated. Mandatory. Once specified, cannot be changed.
- **physSwitchDBID** — A unique identifier of the **Physical Switch** within which this switch is defined. Mandatory. Once specified, cannot be changed.
- **type** — Type of the physical switch to which this switch belongs. Read-only (set automatically according to the current value of type of the physical switch specified in physSwitchDBID). See **CfgSwitchType**.
- **name** — A pointer to the name of the switch. Mandatory. Must be unique within the tenant and the physical switch.
- **TServerDBID** — A unique identifier of the T-Server **Application** through which the telephony objects of this switch are controlled. Parameter tenantDBIDs of the T-Server must be specified and match the setting of tenantDBID of this switch. One T-Server cannot be associated with more than one switch

unless the switch is of type `CFGMultimediaSwitch`. The property is applicable for 5.1 applications only, for compatibility.

Starting from release 6.0 the association between T-Server and switch have to be configured using `CfgApplication (T-Server)` object. See `flexibleProperties` in [CfgApplication](#).

- `linkType` — Type of the CTI link of this switch. Optional. See [CfgLinkType](#).
- `switchAccessCodes` — A pointer to the list of access codes of the switches that this switch can access (every item of this list is structured as [CfgSwitchAccessCode](#)).
When used as an entry in [CfgDeltaSwitch](#), it is a pointer to a list of switch access codes added to the existing list.
- `DNRange` — A pointer to a string that describes the numbering plan of the switch. Use a hyphen to specify a range of numbers; use commas to specify a series of stand-alone numbers or ranges (e.g., 1100-1179, 1190-1195, 1199).
- `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

The current version of Configuration Server does not verify correspondence between the switch numbering plan defined by `DNRange` and the actual DN numbers defined within this switch by number in [CfgDN](#). Such verification may be implemented in one of the next versions. In versions 5.1.1XX and earlier, it is users' responsibility to make sure the range covers all actual DNs of the switch in question.

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

- Modification of `switchAccessCodes` of all switches that were interfaced with Switch X
- Modification of voice prompt objects which had `switchDBID` field set to Switch X
- Modification of `accessNumbers` of all DNs which were connected with Switch X
- Modifications of `flexibleProperties` of all T-Servers which were connected with Switch X
- Deletion of all DNs of Switch X (see comments to [CfgDN](#) for details)
- Deletion of all agent logins of Switch X (see comments to [CfgAgentLogin](#) for details)
- Deletion of all folders that had Switch X defined as the parent object
- Deletion of Switch X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgSwitch>
  <DBID value="101" />
  <tenantDBID value="1" />
  <physSwitchDBID value="101" />
  <type value="12" />
  <name value="Switch" />
  <TServerDBID value="0" />
  <linkType value="0" />
  <state value="1" />
</CfgSwitch>
```

See Also

[CfgDeltaSwitch](#)

[CfgDN](#)

[CfgAgentLogin](#)

CfgSwitchAccessCode

Description

CfgSwitchAccessCode contains a list of Access Codes that are used to place, route, or transfer calls from its Switch to other Switches in a multi-site installation.

Depending on the structure of a numbering plan, you may or may not need access codes to reach DNs that belong to different Switches of a multi-site telephone network.

You can modify (that is, create, change, or delete) the contents of the Access Codes for a particular Switch or for a set of Switches.

Attributes

- `switchDBID` — A unique identifier of the [Switch](#) to which this access code is assigned. Mandatory. If value is set to 0 the `accessCode` value is used as default access to this switch if no other access code is specified on source switch to access this switch.
- `accessCode` — A pointer to the access code.
- `targetType` — Type of the target within the switch specified by `switchDBID` for which all the routing parameters below are specified. See [CfgTargetType](#).
- `routeType` — Type of routing for the target specified in `targetType` for this switch. See [CfgRouteType](#).
- `dnSource` — Source of information to specify parameter `dn` in function `TRouteCall`. See comments.
- `destinationSource` — Source of information to specify parameter `destination` in function `TRouteCall`. See comments.
- `locationSource` — Source of information to specify parameter `location` in function `TRouteCall`.
- `dnisSource` — Source of information to specify parameter `dnis` in function `TRouteCall`.
- `reasonSource` — Source of information to specify parameter `reasons` in function `TRouteCall`.
- `extensionSource` — Source of information to specify parameter `extensions` in function `TRouteCall`.

Comments

Uniqueness of a switch access code is defined by the combination of values of its first three properties, i.e., `switchDBID`, `accessCode`, and `targetType`. Thus, when a certain access code is to be deleted, it is necessary and sufficient to specify those three parameters in the corresponding item of the `deletedSwitchAccessCodes` list in [CfgDeltaSwitch](#).

Function `TRouteCall` is a function of the T-Library and is defined in the T-Library SDK C Developer's

Guide.

See also [CfgSwitch](#).

If targetType=CFGTargetISCC the dnSource property is used for definition of ISCC protocol parameters and presented on GUI (Configuration Manager) with caption "ISCC Protocol Parameters".

If targetType=CFGTargetISCC the destinationSource property is used for definition of ISCC call overflow parameters and presented on GUI (Configuration Manager) with caption "ISCC Call Overflow Parameters".

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 CfgTableType). 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 CfgObjectState). 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](#)

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 CfgObjectState). 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           = CFGEnabled
```

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>
  <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](#)

CfgTimeZone

Description

Time Zones are predefined objects that provide CTI applications with information about world time zones. Each object describes one time zone.

Filter Keys

| Filter Name | Type | Description |
|-----------------|--------|--|
| dbid | int | A unique identifier of the time zone. If specified, Configuration Server will return information only about this time zone. |
| name | string | Name of a time zone. Shall be specified as a character string. If specified, Configuration Server will return information only about the time zone with that name. |
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the time zone(s) that belong to this tenant. |
| offset | int | A time zone offset. If specified, Configuration Server will return information only about the time zone(s) with that offset. |
| name_netscape | string | A pointer to the time zone name used by Netscape Navigator browser. Shall be specified as a character string. If specified, Configuration Server will return information only about the time zone(s) with that name. |
| name_msexplorer | string | A pointer to the time zone name used by Microsoft browser. Shall be specified as a character string. If specified, Configuration Server will return information only about the time zone(s) with that name. |
| state | int | Current state of the time zone (see CfgObjectState). If specified, |

| Filter Name | Type | Description |
|-------------|------|--|
| | | Configuration Server will return information only about the time zone(s) 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 time zone is allocated. Mandatory. Once specified, cannot be changed.
- **name** — A pointer to time zone name. Mandatory. Must be unique within tenant environment.
- **description** — A pointer to the time zone description.
- **offset** — A time zone offset. Any integer value from -24 to 24. Must be considered as value -12 to 12 with 0.5 hour step.
- **isDSTobserved** — A flag which determines whether or not DST is used. Refer to **CfgFlag** from User Defined Variable Types.
- **DSTStartDate** — DST start date. The value is: measured in seconds if 6.0 definition schema is uses. Refer to `time_t` from `time.h` of ANSI C library. Year value range 0-2038; performed based on calculation schema. (See comments.)
- **DSTStopDate** — DST stop date. The value is: measured in seconds if 6.0 definition schema is uses. Refer to `time_t` from `time.h` of ANSI C library. Year value range 0-2038; performed based on calculation schema. (See comments.)
- **DSTOffset** — The value of DST offset. Default is 60 (minutes).
- **nameNetscape** — A pointer to the time zone name used by Netscape Navigator browser. Mandatory.
- **nameMSExplorer** — A pointer to the time zone name used by Microsoft browser. Mandatory
- **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

`time_t` = `int` or `long` (i.e. at least a 32-bit value)

The `time_t` type is logically divided into several sections that contain important information, as described by the following bit-mask:

Time_t Bit Mask Description

| Bits | Description | Range | Comments |
|-------|------------------------|-------------------------|---|
| 0-3 | Month | 0-12 | <ul style="list-style-type: none"> DST is not Observed=0 Jan=1 ... Dec=12 |
| ##### | Week | 0-5, 7 | <ul style="list-style-type: none"> DST is not observed or week is not specified = 0 Last week of month = 7 <p>Note:</p> <ul style="list-style-type: none"> The day of last week of month=week will be set to 7, if the day of week does not occur on last week The day of last week of month=week will be set to 1, if the day of week does not occur on first week |
| ##### | Day | 0-31, 63 | <ul style="list-style-type: none"> DST is not observed = 0 Last day of month = 63 If week is specified (week!=0) the range should be 1-7 Sun=1 ... Sat=7 |
| 13-18 | Start_time, Stop_Time | 0-47 in 30 minute units | 1:00 am = 2 |
| 19-24 | Year (shift from 2000) | 0, 1-38, 39-63 | <ul style="list-style-type: none"> Only if Time Zone is defined for specific Year. 2001 = 1 DST is not observed or year is not specified=0 Values within range 39-63 are not valid |

| Bits | Description | Range | Comments |
|-------|--|----------|--|
| 25-30 | reserved | reserved | reserved |
| ##### | A flag to recognize custom/6.0 time zone | reserved | <ul style="list-style-type: none"> Has to be used to distinguish custom time zones and time zones created before release 6.1: New Style = 1 Old Style or custom time zone = 0 |

Definition of Time Zones for Calculation Schema

| Name | Description | offset | IsDST | Month | Week | Date | Time_start | Month | Week | Date | Time_stop | Year |
|------|------------------------------|--------|-------|-------|------|------|------------|-------|------|------|-----------|------|
| GMT | Greenwich Mean Time | 0 | TRUE | 3 | 7 | 1 | 4 | 10 | 7 | 1 | 6 | 0 |
| ECT | European Central Time | 2 | TRUE | 3 | 7 | 1 | 4 | 10 | 7 | 1 | 6 | 0 |
| EET | Eastern European Time | 4 | TRUE | 3 | 7 | 1 | 6 | 10 | 7 | 1 | 8 | 0 |
| ART | (Arabic) Egypt Standard Time | 4 | TRUE | 4 | 7 | 6 | 0 | 9 | 7 | 6 | 0 | 0 |
| EAT | Eastern African Time | 6 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MET | Middle East Time | 7 | TRUE | 3 | 0 | 20 | 0 | 9 | 0 | 22 | 0 | 0 |
| NET | Near East Time | 8 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PLT | Pakistan Lahore Time | 10 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| IST | India Standard Time | 11 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| BST | Bangladesh Standard Time | 12 | TRUE | 3 | 7 | 1 | 0 | 10 | 7 | 1 | 0 | 0 |
| VST | Vietnam Standard | 14 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Name | Description | Offset | IsDST | Month | Week | Date | Time_start | Month | Week | Date | Time_stop | Year |
|------|---------------------------|--------|-------|-------|------|------|------------|-------|------|------|-----------|------|
| | Time | | | | | | | | | | | |
| CTT | China Taiwan Time | 16 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| JST | Japan Standard Time | 18 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| KST | Korea Standard Time | 18 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ACT | Australia Central Time | 19 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| AET | Australia Eastern Time | 20 | TRUE | 8 | 7 | 7 | 4 | 3 | 7 | 1 | 4 | 0 |
| SST | Solomon Standard Time | 22 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NST | New Zealand Standard Time | 24 | TRUE | 10 | 1 | 1 | 4 | 3 | 3 | 1 | 6 | 0 |
| MIT | Midway Islands Time | -22 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| HST | Hawaii Standard Time | 20 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| AST | Alaska Standard Time | 18 | TRUE | 4 | 1 | 1 | 4 | 10 | 7 | 1 | 4 | 0 |
| PST | Pacific Standard Time | 16 | TRUE | 4 | 1 | 1 | 4 | 10 | 7 | 1 | 4 | 0 |
| PNT | Phoenix Standard Time | 14 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MST | Mountain Standard Time | 14 | TRUE | 4 | 1 | 1 | 4 | 10 | 7 | 1 | 4 | 0 |
| CST | Central Standard Time | 12 | TRUE | 4 | 1 | 1 | 4 | 10 | 7 | 1 | 4 | 0 |
| EST | Eastern Standard Time | 10 | TRUE | 4 | 1 | 1 | 4 | 10 | 7 | 1 | 4 | 0 |
| IET | Indiana | -10 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Name | Description | Offset | IsDST | Month | Week | Date | Time_start | Month | Week | Date | Time_stop | Year |
|-------|--|--------|-------|-------|------|------|------------|-------|------|------|-----------|------|
| | Eastern Standard | | | | | | | | | | | |
| PRT | Puerto Rico and US Virgin Islands Time | -8 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CNT | Canada Newfoundland Time | -7 | TRUE | 4 | 1 | 1 | 4 | 10 | 7 | 1 | 4 | 0 |
| AGT | Argentina Standard Time | -6 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| BET | Brazil Eastern Time | -6 | TRUE | 10 | 2 | 1 | 0 | 2 | 7 | 1 | 0 | 0 |
| CAT | Central African Time | -2 | FALSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| AtIST | Atlantic Standard Time | -8 | TRUE | 4 | 1 | 1 | 4 | 10 | 7 | 1 | 4 | 0 |

The DSTStartTime/DSTStopTime can be calculated using function ConfCalculateTimeZone().

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgTimeZone>
  <DBID value="101" />
  <tenantDBID value="1" />
  <name value="GMT" />
  <description value="Greenwich Mean Time" />
  <offset value="0" />
  <isDSTobserved value="2" />
  <DSTStartDate value="-2147450637" />
  <DSTStopDate value="-2147434246" />
  <nameNetscape value="GMT" />
  <nameMSExplorer value="GMT" />
  <state value="1" />
  <DSTOffset value="60" />
</CfgTimeZone>
```

See Also

[CfgDeltaTimeZone](#)

CfgTransaction

Description

Transactions define how CTI applications calculate customer-defined statistics.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|--|
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the transactions that belong to this tenant. |
| object_type | int | Type of the object (see CfgTransactionType). If specified, Configuration Server will return information only about the transactions of this type. |
| state | int | Current state of a transaction (see CfgObjectState). If specified, Configuration Server will return information only about transactions that are currently in this state. |
| dbid | int | A unique identifier of a transaction. If specified, Configuration Server will return information only about this transaction. |
| name | string | Name of a transaction. Shall be specified as a character string. If specified, Configuration Server will return information only about the transaction(s) with that name. |

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 [Tenant](#) that this transaction belongs to. Mandatory. Once specified, cannot be changed.
- `name` — A pointer to the name of the transaction. Mandatory. Must be unique within the transaction type specified below for the given tenant.
- `type` — Type of the transaction. See type [CfgTansactionType](#). Mandatory. Once specified, cannot be changed. Shall be set to `CFGTransaction` for PEG counts.
- `recordPeriod` — Period in minutes that shows how often the current value of the transaction is to be reported or recorded in a data storage.
- `alias` — An alternative name for this transaction. If specified, must be unique within the object type specified above for the given tenant .
- `description` — A pointer to the text description of the transaction.
- `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.

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgTransaction>
  <DBID value="101" />
  <tenantDBID value="101" />
  <name value="Transaction" />
  <type value="20" />
  <recordPeriod value="0" />
  <alias value="Transaction" />
  <description value="Transaction" />
  <state value="1" />
  <userProperties>
    <list_pair key="Properly">
      <str_pair key="key" value="value" />
    </list_pair>
  </userProperties>
</CfgTransaction>
```

See Also

[CfgDeltaTransaction](#)

CfgTreatment

Description

Treatments, which are most often used in automated outbound campaigns, tell Outbound Contact Server (OCS) how to respond to an unsuccessful call result (a call that does not reach the intended party). For example, the response to an unsuccessful connection may be to redial, and the response to a successful connection may be to play a message.

A *Treatment Sequence* is a group of treatment objects that all contain the same Call Result value. Treatments in a sequence are applied to a call in their numerical order (see the Treatment property Number in Sequence).

Filter Keys

| Filter Name | Type | Description |
|-----------------|--------|--|
| dbid | int | A unique identifier of the treatment. If specified, configuration server will return information only about this treatment. |
| tenant_dbid | int | A unique identifier of the tenant. If specified, Configuration server will return information only about the treatment(s) that belong to this tenant. |
| name | string | Name of a treatment. Shall be specified as a character string. If specified, Configuration Server will return information only about the treatment(s) with that name. |
| call_result | int | A call result related to this treatment (see GctiCallState). If specified, Configuration Server will return information only about the treatments(s) with that call result. |
| rec_action_code | int | A record action code (see CfgRecActionCode). If specified, Configuration Server will return information only about the treatments(s) with that record action code. |
| dest_dn_dbid | int | A unique identifier of destination |

| Filter Name | Type | Description |
|------------------|------|--|
| | | dn. If specified, Configuration Server will return information only about the treatments(s) with that destination dn specified. |
| call_action_code | int | A call action code (see CfgCallActionCode). If specified, Configuration Server will return information only about the treatments(s) with that call action code. |
| state | int | Current state of the table access (see CfgObjectState). If specified, Configuration Server will return information only about the table access(s) 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 treatment action is allocated. Mandatory. Once specified, cannot be changed.
- name — A pointer to treatment action name. Mandatory.
- description — A pointer to treatment action description.
- callResult — A call result related to this treatment. Refer to [GctiCallState](#) in Variable Types of Common APIs. Mandatory.
- recActionCode — A record action code. Refer to [CfgRecActionCode](#) in User Defined Variable Types. See the table under [CfgRecActionCode and Treatment Attributes](#).
- dateTime — A time and date when another attempt must be applied again to dn. The parameter is used if recActionCode is set to CFGRACRetryAtDate. Refer to time_t from time.h of ANSI C library.
- cycleAttempt — An maximum number of sequential attempts the treatment can be applied to dn. The parameter is used if recActionCode is set to CFGRACCycle.
- interval — A time interval in minutes between attempts. The parameter is used if recActionCode is set either to CFGRACCycle or CFGRACRetryIn.
- increment — The time in interval in minutes which increments the interval after each attempt. The parameter is used if recActionCode is set to CFGRACCycle.
- callActionCode — A call action code. Refer to [CfgCallActionCode](#) in User Defined Variable Types. The callActionCode can be applied to following call results only. (Refer to GctiCallState in Variable Types of Common APIs):
 - GctiCStAnswMachine
 - GctiCStFaxDetected

- GctiCStPagerDetected
- GctiCStAnswer
- destDnDBID — A unique identifier of the dn to which the call with this call result will be forwarded/routed. The dns of following types can be used to specify this parameter:
 - CFGExtension
 - CFGACDPosition
 - CFGACDQueue
 - CFGRoutingPoint
 - CFGEAPort
 - CFGVoiceMail
 - CFGFAX
 - CFGMusic

Refer to CfgDNTypes of User Defined Variable Types.

This parameter is recommended to be defined for following callActionCodes:

 - CFGCACMuteTransfer
 - CFGCACTransfer
 - CFGCACRoute

Refer also to [CfgCallActionCode](#) of User Defined Variable Types.
- attempts — An attempt number to which the action should be performed.
- 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.
- range — Parameter defining a time range

CfgRecActionCode and Treatment Attributes

The table below contains relationship between [CfgRecActionCode](#) and treatment attributes. Attributes marked with * are mandatory. Defaults are specified for GUI representation only. Parameters marked with N/A (Not Applicable) must be disabled in Configuration Manager GUI for this action code.

| CFGRecActionCode | attempts | dateTime | cycle-Attempt | interval | increment | destDnDBID |
|----------------------|-----------|----------|---------------|------------|------------|------------|
| | range:1-n | | range: 0-n | range: 1-n | range: 0-n | |
| CFGGRAC-MarkDB | * | N/A | N/A | N/A | N/A | |
| CFGGRAC-MarkAllChain | * | N/A | N/A | N/A | N/A | N/A |

| CFGRecActionCode | cycle-Attempt | dateTime | cycle-Attempt | interval | increment | destDNDBID |
|--------------------------|---------------|----------|--|--------------------------------------|-------------------------------|------------|
| CFGGRAC-Cycle | * | N/A | * | * | * | N/A |
| | | | Default is 10 | Default is 1 | Default is 0 | |
| | | | Caption: "Number of attempts" | Caption: "Interval between attempts" | Caption: "Increment interval" | |
| CFGGRAC-RetryIn | * | N/A | N/A | * | N/A | |
| | | | | Default is 30 | | |
| | | | | Caption: "Retry in minutes" | | |
| CFGGRAC-RetryAtDate | * | * | N/A | N/A | N/A | |
| CFGGRACNextInChain | | N/A | * = Chain recycle if >0 | N/A | N/A | N/A |
| | | | Default is 0 | | | |
| | | | Caption: "Chain recycle" can be done as checkbox | | | |
| CFGGRACNextInChainAfter | | N/A | * = Chain recycle if >0 | * | N/A | N/A |
| | | | Default is 0 | Default is 30 | | |
| | | | Caption: "Chain recycle" can be done as checkbox | Caption: "Retry in minutes" | | |
| CFGGRACNextInChainAtDate | * | * | * = Chain recycle if >0 | N/A | N/A | N/A |
| | | | Default is 0 | | | |
| | | | Caption: "Chain recycle" can be done as checkbox | | | |
| CFGGRACAssignToGroup | N/A | N/A | N/A | N/A | N/A | N/A |
| CFGGRACMarkAsAgentError | N/A | N/A | N/A | N/A | N/A | N/A |
| CFGGRACReschedule | * | * | * | * | * | N/A |

If the `GctiCallState = GctiCStAgentCallBackErr`, the following types of `CFGRecActionCode` could be used within the `recActionCode` property only:

- CFGRACAssignToGroup
- CFGRACMarkAsAgentError
- CFGRACReschedule
- CFGRACCycle ("Redial")
- CFGRACRetryIn ("Retry in")
- CFGRACRetryAtDate ("Retry at specified date")

Comments

One Treatment can be associated with several CallingLists (see [CfgCallingList](#)).

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

- Modification of treatmentDBIDs of all calling lists that included Treatment X
- Deletion of Treatment X

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgTreatment>
  <DBID value="101" />
  <tenantDBID value="101" />
  <name value="Test_Treatment" />
  <description value="test treatment" />
  <callResult value="0" />
  <recActionCode value="6" />
  <attempts value="1" />
  <dateTime value="1170292126" />
  <cycleAttempt value="0" />
  <interval value="0" />
  <increment value="0" />
  <callActionCode value="9" />
  <destDNDBID value="0" />
  <state value="1" />
  <range value="0" />
</CfgTreatment>
```

See Also

[CfgDeltaTreatment](#)

[CfgCallingList](#)

CfgVoicePrompt

Description

Voice Prompts are call treatment objects that may include a set of actions to be applied to a called party.

Filter Keys

| Filter Name | Type | Description |
|-------------|--------|--|
| dbid | int | A unique identifier of a voice prompt. If specified, Configuration Server will return information only about this voice prompt. |
| name | string | Name of a voice prompt. Shall be specified as a character string. If specified, Configuration Server will return information only about the voice prompts(s) with that name. |
| tenant_dbid | int | A unique identifier of a tenant. If specified, Configuration Server will return information only about the voice prompts that belong to this tenant. |
| switch_dbid | int | A unique identifier of a switch. If specified, Configuration Server will return information only about the voice prompts that belong to this switch. |
| script_dbid | int | A unique identifier of a script. If specified, Configuration Server will return information only about the voice prompts that refer to this script. |
| state | int | Current state of a voice prompt (see CfgObjectState). If specified, Configuration Server will return information only about voice prompts 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.
- switchDBID — A unique identifier of the [CfgSwitch](#) to which this voice prompt belongs. Optional.
- tenantDBID — A unique identifier of the [CfgTenant](#) to which this voice prompt is allocated. Read only.
- name — A pointer to the voice prompt name that identifies the voice prompt. Should be equal to the voice prompt identifier provided by switch if used for parked call treating. Mandatory. Once specified cannot be changed. Must be unique within tenant it belongs to.
- description — A pointer to the voice prompt description. Can be used by parking platform or switch for generating voice prompt (Text-to-Speech).
- scriptDBID — A unique identifier of the [CfgScript](#) for this voice prompt.
- 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

Common/System Voice Prompts have to be located under \Environment \ Voice Prompts folders. The System Voice Prompts is available for all applications.

Tenant-specific Voice Prompts have to be located under <Tenant>\ Voice Prompts folders. The access to these prompts have only configuration objects with access permissions to this tenant only.

XML Representation

Tip

This XML was created using the Configuration Server 7.5 schema.

```
<CfgVoicePrompt>
  <DBID value="101" />
  <switchDBID value="113" />
  <tenantDBID value="101" />
  <name value="Test_Voice_Prompt" />
  <scriptDBID value="0" />
  <state value="1" />
</CfgVoicePrompt>
```

See Also

[CfgDeltaVoicePrompt](#)

List of Configuration Layer Enumerations

The following table provides a convenient list of Configuration Layer Enumerations that are available. For more information, refer to [Introduction to the Configuration Layer Objects](#).

| | | | |
|-------------------------|-------------------|-----------------------|--------------------|
| CfgAccessGroupType | CfgEnumeratorType | CfgOSType | CfgSolutionType |
| CfgActionCodeType | CfgErrorType | CfgObjectState | CfgStartupType |
| CfgAlarmCategory | CfgEventType | CfgObjectType | CfgStatDayType |
| CfgAppComponentType | CfgFieldType | CfgObjectiveTableType | CfgStatTableType |
| CfgAppType | CfgFilterType | CfgOperationMode | CfgSwitchType |
| CfgCallActionCode | CfgFlag | CfgOperationalMode | CfgTableType |
| CfgChargeType | CfgFolderClass | CfgOptimizationMethod | CfgTargetType |
| CfgCTILinkType | CfgGroupType | CfgPermissions | CfgTaskType |
| CfgDIDGroupType | CfgHAType | CfgPersonType | CfgTraceMode |
| CfgDNGroupType | CfgHostType | CfgRank | CfgTransactionType |
| CfgDNRegisterFlag | CfgIVRProfileType | CfgRecActionCode | GctiCallState |
| CfgDNType | CfgIVRType | CfgResourceType | GctiContactType |
| CfgDataType | CfgLanguage | CfgRouteType | GctiRecordStatus |
| CfgDialMode | CfgLinkType | CfgScriptType | GctiRecordType |
| CfgEnumeratorObjectType | CfgMediaType | CfgSelectionMode | |

CfgAccessGroupType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------------------|----|------|-----------------------|---------------------|------------------|
| CFGNoAccessGroupType | 0 | 0 | 'Unknown Type' | NoAccessGroupType | |
| CFGDefaultGroup | 1 | 0 | 'Default Group Type' | DefaultGroupType | |
| CFGUsersGroup | 2 | 0 | 'Users' | Users | |
| CFGAdministratorsGroup | 3 | 0 | 'Administrators' | Administrators | |
| CFGSuperAdministratorsGroup | 4 | 0 | 'SuperAdministrators' | SuperAdministrators | |
| CFGSystemGroup | 5 | 0 | 'SYSTEM' | SYSTEM | |
| CFGEveryoneGroup | 6 | 0 | 'EVERYONE' | EVERYONE | |

CfgActionCodeType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------|----|------|----------------------------|--------------|---|
| CFGNoActionCode0 | | 0 | 'Unknown Action Code Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGInboundCall | 1 | 0 | 'Inbound Call' | InboundCall | |
| CFGOutboundCall | 2 | 0 | 'Outbound Call' | OutboundCall | |
| CFGInternalCall | 3 | 0 | 'Internal Call' | InternalCall | |
| CFGTransfer | 4 | 0 | 'Transfer' | Transfer | |
| CFGConference | 5 | 0 | 'Conference' | Conference | |
| CFGLogin | 6 | 0 | 'Login' | Login | |
| CFGLogout | 7 | 0 | 'Logout' | Logout | |
| CFGReady | 8 | 0 | 'Ready' | Ready | |
| CFGNotReady | 9 | 0 | 'Not Ready' | NotReady | |
| CFGBusyOn | 10 | 0 | 'Busy On' | BusyOn | |
| CFGBusyOff | 11 | 0 | 'Busy Off' | BusyOff | |
| CFGForwardOn | 12 | 0 | 'Forward On' | ForwardOn | |
| CFGForwardOff | 13 | 0 | 'Forward Off' | ForwardOff | |

CfgAlarmCategory

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------|-----------|-------------|--------------------------|-----------------|--|
| CFGACNoAlarmCategory | 0 | 0 | 'Unknown Alarm Category' | Unknown | The value of the corresponding parameter has not changed |
| CFGACCritical | 1 | 0 | 'Critical' | Critical | Critical alarm |
| CFGACMajor | 2 | 0 | 'Major' | Major | Major alarm |
| CFGACMinor | 3 | 0 | 'Minor' | Minor | Minor alarm |

CfgAppComponentType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------------|----|------|-------------------|----------|------------------|
| CFGAppComponentUnknown | 0 | 0 | | | |
| CFGAppComponentSubProcess | 0 | 0 | | | |
| CFGAppComponentService | 0 | 0 | | | |
| CFGAppComponent | 0 | 0 | | | |
| CFGAppComponentAccessPoint | | | | | |

CfgAppType

Values

| Mnemonic | ID | Description |
|---------------------|----|--|
| CFGNoApplication | 0 | The value of the corresponding parameter has not changed. |
| CFGServer | 1 | T-Server |
| CFGStatServer | 2 | Stat Server |
| CFGBillingServer | 3 | Billing System Server |
| CFGBillingClient | 4 | Billing System Client |
| CFGAgentView | 5 | Agent View |
| CFGEAServer | 6 | Electronic Audio/Voice Treatment Server (also known as IVR Server) |
| CFGEAClient | 7 | CFGEAClient constant. |
| CFGDBServer | 8 | DB access point() |
| CFGCallConcentrator | 9 | Call Concentrator |
| CFGSDialer | 10 | S-Dialer |
| CFGListManager | 11 | List Manager |
| CFGCMServer | 12 | Campaign Manager Server |
| CFGCMClient | 13 | Campaign Manager GUI Client Application |
| CFGGLME | 14 | Campaign Configuration Environment (also known as List Management Environment) |
| CFGRouterServer | 15 | ICD Server (also known as Call Router) |
| CFGStrategyBuilder | 16 | ICD Strategy Builder |
| CFGStrategyLoader | 17 | ICD Client (also known as Strategy Loader) |
| CFGAgentDesktop | 18 | Agent Desktop Integrated Package |
| CFGSCCE | 19 | Configuration Manager (formerly known as Service Creator) |
| CFGCCView | 20 | Call Center Pulse (also known as Call Center View) |
| CFGConfigServer | 21 | Configuration Server |
| CFGThirdPartyApp | 22 | A third-party application |

| Mnemonic | ID | Description |
|-------------------------------|-----------|--|
| CFGThirdPartyServer | 23 | A third-party server |
| CFGStrategySimulator | 24 | ICD Strategy Simulator |
| CFGStrategyScheduler | 25 | ICD Strategy Scheduler |
| CFGDARTServer | 26 | DART Server |
| CFGDARTClient | 27 | DART GUI Client Application |
| CFGCustomServer | 28 | Custom Server |
| CFGExternalRouter | 29 | External Router |
| CFGVirtualInteractiveT | 30 | Virtual Interactive T |
| CFGVirtualRP | 31 | Virtual Routing Point |
| CFGDatabase | 32 | A logical database |
| CFGNetVector | 33 | NetVector |
| CFGDetailBiller | 34 | Detailed-Billing System |
| CFGSummaryBiller | 35 | Summary-Billing System |
| CFGNACD | 36 | Network ACD |
| CFGBackUpControlClient | 37 | Back UP Control Client |
| CFGInfomartStatCollector | 38 | Infomart Statistic Collector |
| CFGInfomartStatConfigurator | 39 | Infomart Statistic Configurator |
| CFGIVRInterfaceServer | 40 | IVR Server |
| CFGIServer | 41 | I-Server |
| CFGMessageServer | 42 | Message Server |
| CFGSCS | 43 | Solution Control Server |
| CFGSCI | 44 | Solution Control Interface |
| CFGSNMPAgent | 45 | SNMP Agent |
| CFGRealDBServer | 46 | DB Server |
| CFGWFMClient | 47 | Work Force Management Client |
| CFGWFMDDataAggregator | 48 | Work Force Management Aggregator |
| CFGWFMWebServices | 49 | Work Force Management Web Services |
| CFGWFMScheduleServer | 50 | Work Force Management Schedule Server |
| CFGInteractionRoutingDesigner | 51 | Interaction Routing Designer (Introduced in Genesys 6) |
| CFGETLProxy | 52 | Extract Transform Load Proxy |
| CFGITCUtility | 53 | Install-Time Configuration Utility |
| CFGVCServer | 54 | GVP-Voice Communication Server |
| CFGGDMETL | 55 | Genesys Infomart ETL |
| CFGVSSShared | 56 | Voice Self-Service Shared Application |

| Mnemonic | ID | Description |
|---------------------------------|-----------|---|
| CFGVSSConsole | 57 | Voice Self-Service Console Application |
| CFGCCAnalyzerDataMart | 58 | Contact Center Analyzer Data Mart |
| CFGChatServer | 59 | Chat Server |
| CFGCallbackServer | 60 | Callback Server |
| CFGCoBrowsingServer | 61 | Co-Browsing Server |
| CFGSMSServer | 62 | CFGSMSServer constant. |
| CFGContactServer | 63 | Contact Server |
| CFGEmailServer | 64 | E-mail Server |
| CFGMediaLink | 65 | Media link |
| CFGWebInteractionRequestsServer | 66 | Web Interaction Requests Server |
| CFGWebStatServer | 67 | Web Stat Server |
| CFGWebInteractionServer | 68 | Web Interaction Server |
| CFGWebOptionRoutePoint | 69 | Web Option Route Point |
| CFGWebClient | 70 | Web Client |
| CFGContactServerManager | 71 | Contact Server Manager |
| CFGContentAnalyzer | 72 | Content Analyzer |
| CFGResponseManager | 73 | Response Manager |
| CFGVoIPController | 74 | Voice over IP Controller |
| CFGVoIPDevice | 75 | Voice over IP Device |
| CFGAutomatedWorkflowEngine | 76 | Automated Workflow Engine |
| CFGHAProxy | 77 | Proxy for T-Server High Availability solution |
| CFGVoIPStreamManager | 78 | Voice over IP Stream Manager |
| CFGVoIPDMXServer | 79 | Voice over IP DMX Server |
| CFGWebAPIServer | 80 | Web API Server |
| CFGLoadBalancer | 81 | Load Balancer |
| CFGApplicationCluster | 82 | Application cluster |
| CFGLoadDistributionServer | 83 | Load Distribution Server |
| CFGGProxy | 84 | G-Proxy |
| CFGGIS | 85 | Genesys Interface Server |
| CFGAgentDesktopDeliveryServer | 86 | Genesys Contact Navigator Delivery Server |
| CFGGCNClient | 87 | Genesys Contact Navigator Client |
| CFGIVRDT | 88 | Genesys IVR DirectTalk server |
| CFGGCNThinServer | 89 | GCN Thin server |
| CFGClassificationServer | 90 | Classification Server |
| CFGTrainingServer | 91 | Training Server |

| Mnemonic | ID | Description |
|------------------------------|-----------|--|
| CFGUniversalCallbackServer | 92 | Universal Callback Server |
| CFGCPDServerProxy | 93 | CPD Server Proxy |
| CFGXLinkController | 94 | XLink Controller |
| CFGKWorkerPortal | 95 | Knowledge Worker Portal |
| CFGWFMServer | 96 | WFM Server |
| CFGWFMBuilder | 97 | WFM Builder |
| CFGWFMSReports | 98 | WFM Reports |
| CFGWFMSWeb | 99 | WFM Web |
| CFGKnowledgeManager | 100 | Knowledge Manager |
| CFGIVRDriver | 101 | IVR Driver |
| CFGIVRLibrary | 102 | IVR Library |
| CFGLCAdapter | 103 | LCS Adapter |
| CFGDesktopNETServer | 104 | Desktop .NET Server |
| CFGSiebel7ConfSynchComponent | 105 | Gplus Adapter for Siebel 7 Configuration Synchronization Component |
| CFGSiebel7CampSynchComponent | 106 | Gplus Adapter for Siebel 7 Campaign Synchronization Component |
| CFGGenericServer | 107 | Genesys Generic Server |
| CFGGenericClient | 108 | Genesys Generic Client |
| CFGCallDirector | 109 | Call Director |
| CFGSIPCommunicationServer | 110 | SIP Communication Server |
| CFGInteractionServer | 111 | Interaction Server |
| CFGIntegrationServer | 112 | CFGIntegrationServer constant. |
| CFGWFMDaemon | 113 | CFGWFMDaemon constant. |
| CFGGVPPolicyManager | 114 | CFGGVPPolicyManager constant. |
| CFGGVPCiscoQueueAdapter | 115 | CFGGVPCiscoQueueAdapter constant. |
| CFGGVPTextToSpeechServer | 116 | CFGGVPTextToSpeechServer constant. |
| CFGGVPPASRLogManager | 117 | CFGGVPPASRLogManager constant. |
| CFGGVPSBandwidthManager | 118 | CFGGVPSBandwidthManager constant. |
| CFGGVPEventsCollector | 119 | CFGGVPEventsCollector constant. |
| CFGGVPCacheServer | 120 | CFGGVPCacheServer constant. |
| CFGGVPPASRLogServer | 121 | CFGGVPPASRLogServer constant. |
| CFGGVPPASRPackageLoader | 122 | CFGGVPPASRPackageLoader constant. |

| Mnemonic | ID | Description |
|-------------------------------------|-----|---|
| CFGGVPIPCommunicationServer | 123 | CFGGVPIPCommunicationServer constant. |
| CFGGVPResourceManager | 124 | CFGGVPResourceManager constant. |
| CFGGVPSIPSessionManager | 125 | CFGGVPSIPSessionManager constant. |
| CFGGVPMediaGateway | 126 | CFGGVPMediaGateway constant. |
| CFGGVPSoftSwitch | 127 | CFGGVPSoftSwitch constant. |
| CFGGVPCoreService | 128 | CFGGVPCoreService constant. |
| CFGGVPVoiceCommunicationServer | 129 | CFGGVPVoiceCommunicationServer constant. |
| CFGGVPUnifiedLoginServer | 130 | CFGGVPUnifiedLoginServer constant. |
| CFGGVPCallStatusMonitor | 131 | CFGGVPCallStatusMonitor constant. |
| CFGGVPReporter | 132 | CFGGVPReporter constant. |
| CFGGVPH323SessionManager | 133 | CFGGVPH323SessionManager constant. |
| CFGGVPASRLogManagerAgent | 134 | CFGGVPASRLogManagerAgent constant. |
| CFGGVPGenesysQueueAdapter | 135 | CFGGVPGenesysQueueAdapter constant. |
| CFGGVPIServer | 136 | CFGGVPIServer constant. |
| CFGGVPSCPGateway | 137 | CFGGVPSCPGateway constant. |
| CFGGVPSRPServer | 138 | CFGGVPSRPServer constant. |
| CFGGVPMRCPTTServer | 139 | CFGGVPMRCPTTServer constant. |
| CFGGVPCCSServer | 140 | CFGGVPCCSServer constant. |
| CFGGVPMRCPASRServer | 141 | CFGGVPMRCPASRServer constant. |
| CFGGVPNetworkMonitor | 142 | CFGGVPNetworkMonitor constant. |
| CFGGVPOBNManager | 143 | CFGGVPOBNManager constant. |
| CFGGVPSelfServiceProvisioningServer | 144 | CFGGVPSelfServiceProvisioningServer constant. |
| CFGGVPMCP | 145 | GVP Media Control Platform |
| CFGGVPFetchingModule | 146 | GVP Fetching Module |
| CFGGVPMCPLegacyInterpreter | 147 | GVP Media Control Platform Legacy Interpreter |
| CFGGVPCCP | 148 | GVP Call Control Platform |
| CFGGVPResourceMgr | 149 | GVP Resource Manager |
| CFGGVPClusterMgr | 150 | CFGGVPClusterMgr constant. |

| Mnemonic | ID | Description |
|-----------------------------------|-----------|-----------------------------------|
| CFGGVPMediaServer | 151 | GVP Media Server |
| CFGGVPPSTNConnector | 152 | GVP PSTN Connector |
| CFGGVPReportingSever | 153 | GVP Reporting Server |
| CFGGVPPASG | 154 | GVP ASG |
| CFGGVPPCTIConnector | 155 | GVP CTI Connector |
| CFGResourceAccessPoint | 156 | CFGResourceAccessPoint constant. |
| CFGInteractionWorkspace | 157 | Interaction Workspace |
| CFGAdvisors | 158 | Advisors |
| CFGESSExtensibleServices | 159 | ESS Extensible Services |
| CFGCustomerView | 160 | Customer View |
| CFGOrchestrationServer | 161 | Orchestration Server |
| CFGReserved | 162 | Reserved |
| CFGCapturePoint | 163 | Capture Point |
| CFGRulesESPServer | 164 | Rules ESP Server |
| CFGGenesysAdministrator | 165 | Genesys Administrator |
| CFGiWDManager | 166 | iWD Manager |
| CFGiWDRuntimeNode | 167 | iWD Runtime Node |
| CFGBusinessRulesExecutionServer | 168 | Business Rules Execution Server |
| CFGBusinessRulesApplicationServer | 169 | Business Rules Application Server |
| CFGVPPolicyServer | 170 | VP Policy Server |
| CFGSocialIMS | 171 | Social Messaging Server |
| CFGCSTACConnector | 172 | CSTA Connector |
| CFGVPMRCPProxy | 173 | VP MRCP Proxy |
| CFGUCMConnector | 174 | UCM Connector |
| CFGOTICSServer | 175 | OT ICS Server |
| CFGOTICSOMPIinfra | 176 | OT ICS OMPInfra |
| CFGCCACCAdviser | 177 | Advisors-Contact Center Advisor |
| CFGCCAFAAdviser | 178 | Advisors-Frontline Advisor |
| CFGCCAPIatform | 179 | Advisors-Advisors Platform |
| CFGCCAGenesysAdapter | 180 | Advisors-Advisors Genesys Adapter |
| CFGCCACiscoAdapter | 181 | Advisors-Advisors Cisco Adapter |
| CFGFederationServer | 182 | Federation Server |
| CFGFederationStatProvider | 183 | Federation Stat Provider |
| CFGGenesysAdministratorServer | 184 | Genesys Administrator Server |
| CFGWebEngagementBackendServer | 185 | Web Engagement Backend Server |

| Mnemonic | ID | Description |
|--------------------------------|-----------|---|
| CFGWebEngagementFrontEndServer | 186 | Web Engagement Frontend Server |
| CFGWebRTCGateway | 187 | CFGWebRTCGateway constant. |
| CFGLRMServer | 188 | CFGLRMServer constant. |
| CFGRecordingCryptoServer | 189 | CFGRecordingCryptoServer constant. |
| CFGKnowledgeCenter | 190 | CFGKnowledgeCenter constant. |
| CFGKnowledgeCenterCMS | 191 | CFGKnowledgeCenterCMS constant. |
| CFGReservedApplication8 | 192 | Reserved Application 8 |
| CFGReservedApplication9 | 193 | Reserved Application 9 |
| CFGReservedApplication10 | 194 | Reserved Application 10 |
| CFGReservedGUIApplication1 | 195 | CFGReservedGUIApplication1 constant. |
| CFGReservedGUIApplication2 | 196 | CFGReservedGUIApplication2 constant. |
| CFGMaxAppType | 197 | A maximum value of this enumeration type. |

CfgCallActionCode

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------------|----|------|-----------------------|------------------|--|
| CFGCACNoCallActionCode | 0 | 0 | 'Unknown Action Code' | NoCallActionCode | The value of corresponding parameter has not changed |
| CFGCACConnect | 1 | 0 | 'Connect' | Connect | Connect to an operator |
| CFGCACDrop | 2 | 0 | 'Drop' | Drop | Release call |
| CFGCACMuteTransfer | 3 | 0 | 'Mute Transfer' | MuteTransfer | Transfer call using mute transfer |
| CFGCACTransfer | 4 | 0 | 'Transfer' | Transfer | Transfer call using two-step transfer |
| CFGCACRoute | 5 | 0 | 'Route' | Route | Route call |
| CFGCACPlayMessage | 6 | 0 | 'Play a message' | PlayMessage | Slay message |
| CFGCACSendFax | 7 | 0 | 'Send a fax' | SendFax | Send fax |
| CFGCACSendPage | 8 | 0 | 'Send a page' | SendPage | Send page |
| CFGCACSendEmail | 9 | 0 | 'Send an e-mail' | SendEmail | Send E-mail |

Comments

CAC in value name means belonging to CfgCallActionCode enumeration.

The CfgCallActionCode is applicable for Configuration Library/Server release 5.1.5xx only.

CfgChargeType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------|----|------|-----------------------|----------|---|
| CFGNoChargeType0 | | 0 | 'Unknown Charge Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGFree | 1 | 0 | 'Free' | Free | The service is provided free of charge. |
| CFGFlatRate | 2 | 0 | 'Flat Rate' | FlatRate | Flat-rate charging. |
| CFGInstance | 3 | 0 | 'Instance' | Instance | Instance-based charging. |
| CFGDuration | 4 | 0 | 'Duration' | Duration | Duration-based charging. |

CfgCTILinkType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------|-----------|-------------|--------------------------|-----------------|-------------------------|
| CFGUndefined | 0 | 0 | | | |
| CFGTCP | 1 | 0 | | | |
| CFGX25 | 2 | 0 | | | |
| CFGVendorLibrary3 | | 0 | | | |

CfgDIDGroupType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|-----------------|-----------|-------------|--------------------------|-----------------|
| CFGDGType | 0 | 0 | 'Unknown' | Unknown |
| CFGDGDialable | 1 | 0 | 'Dialable' | Dialable |
| CFGDGReRoute | 2 | 0 | 'ReRoute' | ReRoute |

CfgDNGroupType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------|----|------|----------------------|---------------|--|
| CFGNoDNGroup | 0 | 0 | 'Unknown Group Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGSinglePorts | 1 | 0 | 'Single Ports' | SinglePorts | Can include DNs of the following types: CFGExtension, CFGACDPosition, CFGGEAPort, CFGVoiceMail, CFGCP, CFGMusic, CFGData, CFGFAX, CFGCommDN, CFGEmal, CFGVoIP, CFGVideo, CFGChat, CFGCoBrowse. |
| CFGACDQueues | 2 | 0 | 'ACD Queues' | ACDQueues | Can only include DNs of CFGACDQueue, CFGVirtACDQueue, and CFGRoutingQueue type. |
| CFGRoutingPoints | 3 | 0 | 'Routing Points' | RoutingPoints | Can include DNs of the following types: CFGRoutingPoint, CFGExtRoutingPoint, CFGVirtRoutingPoint, and CFGRoutingQueue. |
| CFGNetworkPorts | 4 | 0 | 'Network Ports' | NetworkPorts | Can only include DNs of CFGDestinationLabel type. |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------|-----------|-------------|--------------------------|-----------------|--|
| CFGServiceNumbers | 5 | 0 | 'Service Numbers' | ServiceNumbers | Can only include DNs of CFGServiceNumber type. |

CfgDNRegisterFlag

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------|-----------|-------------|--------------------------|-----------------|-------------------------|
| CFGDRUnknown | 0 | 0 | 'Unknown Flag' | Unknown | |
| CFGDRFalse | 1 | 0 | 'False' | False | |
| CFGDRTrue | 2 | 0 | 'True' | True | |
| CFGDROnDemand3 | | 0 | 'On Demand' | OnDemand | |

Comments

DR in value name means belonging to CfgDNRegisterFlag enumeration.

CfgDNType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|---------------------|----|------|-------------------------|---------------------|---|
| CFGNoDN | 0 | 0 | 'Unknown DN Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGExtension | 1 | 0 | 'Extension' | Extension | A regular extension line. |
| CFGACDPosition | 2 | 0 | 'ACD Position' | ACDPosition | An extension line designated for customer-agent calls only. |
| CFGACDQueue | 3 | 8 | 'ACD Queue' | ACDQueue | An ACD queue. |
| CFGRoutingPoint | 4 | 8 | 'Routing Point' | RoutingPoint | A routing point (a telephony object within the switch at which calls reside while routing decisions are being made). |
| CFGVirtACDQueue5 | | 8 | 'Virtual Queue' | VirtualACDQueue | A virtual ACD queue. A virtual telephony object handled and reported exclusively by CTI applications and whose event model corresponds to the event model of a regular ACD queue. |
| CFGVirtRoutingPoint | | 8 | 'Virtual Routing Point' | VirtualRoutingPoint | A virtual routing point. A virtual telephony |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------|----|------|------------------------|--------------------|--|
| | | | | | object handled and reported exclusively by CTI applications and whose event model corresponds to the event model of a regular routing point. |
| CFGGEAPort | 7 | 0 | 'Voice Treatment Port' | VoiceTreatmentPort | An extension connected to a electronic audio/voice treatment port (e.g., IVR). |
| CFGVoiceMail | 8 | 0 | 'Voice Mail' | VoiceMail | A voice mail channel. |
| CFGCellular | 9 | 0 | 'Mobile Station' | Cellular | A mobile station. |
| CFGGCP | 10 | 0 | 'Call Processing Port' | CP | An extension connected to a call-processing equipment port. |
| CFGFAX | 11 | 0 | 'Fax' | FAX | An extension connected to a fax machine. |
| CFGData | 12 | 0 | 'Modem' | Data | An extension connected to a data communication equipment. |
| CFGMusic | 13 | 0 | 'Music Port' | Music | A music source. |
| CFGTrunk | 14 | 0 | 'Trunk' | Trunk | A trunk. |
| CFGTrunkGroup | 15 | 0 | 'Trunk Group' | TrunkGroup | A group of trunks forming one route. |
| CFGTieLine | 16 | 0 | 'Tie Line' | TieLine | A tie line. |
| CFGTieLineGroup | 17 | 0 | 'Tie Line Group' | TieLineGroup | A group of tie lines forming one route. |
| CFGMixed | 18 | 0 | 'Mixed' | Mixed | An extension line that can be used as CFGACDPosition |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|---------------------|----|------|--------------------------|------------------|---|
| | | | | | as well as CFGExtension. |
| CFGExtRoutingPoint | 19 | 8 | 'External Routing Point' | ExtRoutingPoint | A routing point that is dedicated to support external routing functions. |
| CFGDestinationLabel | 20 | 0 | 'Network Destination' | DestinationLabel | A destination label used as a destination number in network routing. |
| CFGServiceNumber | 21 | 8 | 'Service Number' | ServiceNumber | A service number used as a routing point in network routing. |
| CFGRoutingQueue | 22 | 8 | 'Routing Queue' | RoutingQueue | A telephony object that has properties of both CFGRoutingPoint and CFGACDQueue. |
| CFGCommDN | 23 | 0 | 'Communication DN' | CommunicationDN | A telephony object that NetVector components use to communicate with a switch. |
| CFGEmail | 24 | 0 | 'E-mail Address' | Email | An e-mail address. |
| CFGVoIP | 25 | 0 | 'Voice over IP Port' | VoIP | A Voice over IP. |
| CFGVideo | 26 | 0 | 'Video over IP Port' | Video | A Video channel. |
| CFGChat | 27 | 0 | 'Chat' | Chat | A Chat address |
| CFGCoBrowse | 28 | 0 | 'CoBrowse' | CoBrowse | A CoBrowse address |
| CFGVoIPService | 29 | 0 | 'Voice over IP Service' | VoIPService | A Voice over IP Service |
| CFGWorkflow | 30 | 0 | 'Workflow' | Workflow | A Workflow DN |
| CFGAccessResource | 31 | 8 | 'Access' | AccessResource | A Switch |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------|----|------|-------------------|----------|--|
| | | | Resource' | | Access Resource to be used in multi-switch environment (ext routing) |
| CFGGVPDID | 32 | 0 | 'GVP DID' | GVPDID | |

CfgDataType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------|----|------|---------------------|------------|---|
| CFGDTNoDataType@ | | 0 | 'Unknown Data Type' | NoDataType | The value of corresponding parameter is not changed. |
| CFGDTInt | 1 | 0 | 'int' | Int | A data base data type that is an integer column that holds whole numbers between 231 -1 (that is, 2,147,483,647) and -231 (that is, -2,147,483,648), inclusive. |
| CFGDTFloat | 2 | 0 | 'float' | Float | A data base data type. It is a floating-point column that holds positive or negative floating-point numbers. This column has 15-digit precision. The range of values is approximately 1.7E - 308 to 1.7E 308. |
| CFGDTChar | 3 | 0 | 'char' | Char | A character data type that holds a maximum of 255 characters. Specify the maximum length with n. The char |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|--------------|----|------|-------------------|----------|---|
| | | | | | datatype can contain 0 characters, but when specified, n must be a value from 1 through 255. Storage size is n regardless of the actual length of the entry. Choose char when you think the data entries in the column will be consistently close to the same size. Columns of type char are accessed somewhat faster than varchar columns because they use a fixed storage length (n). |
| CFGDTVarChar | 4 | 0 | 'varchar' | VarChar | A data base data type, varchar(n) is a column of variable-length characters that holds any combination of up to 255 letters, symbols, and numbers. Specify the maximum size of the column with n. A char column can contain 0 characters, but n must be between 1 and 255, inclusive. Storage size is the actual size |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|---------------|----|------|-------------------|----------|---|
| | | | | | <p>of the data values entered, not n. Because of the way it uses space, varchar is best suited for data-like names, where the length of each entry can vary considerably. But varchar imposes more overhead than char, so if the length is fairly consistent, use char instead. Data of type varchar must be enclosed in single quotation marks when it is input. To have a truly empty sting, insert the keyword NULL rather than an empty string, such as ' '. You can use the LIKE keyword and wildcard characters with varchar. If you enter strings that are too long for the specified length, varchar entries are truncated.</p> |
| CFGDTDateTime | 5 | 0 | 'datetime' | DateTime | <p>A data base data type. A datetime datatype is stored in 8 bytes of two 4-byte integers: 4 bytes for the</p> |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------|----|------|-------------------|----------|---|
| | | | | | number of days before or after the base date of January 1, 1900, and 4 bytes for the number of milliseconds after midnight. The date segments of datetime values representing dates prior to the base date are stored as negative values. |

Comments

An identifier that specifies what type of information a column holds and how the data is stored. The CfgDataType is applicable for Configuration Library/Server release 5.1.5xx only.

CfgDialMode

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------------|----|------|----------------------------|------------------|--|
| CFGDMNoDialMode | 0 | 0 | 'Unknown Dialing Mode' | NoDialMode | The value of corresponding parameter has not changed |
| CFGDMPredict | 1 | 0 | 'Predictive' | Predict | Predictive |
| CFGDMProgress | 2 | 0 | 'Progressive' | Progress | Progressive |
| CFGDMPreview | 3 | 0 | 'Preview' | Preview | Preview |
| CFGDMProgressAndSeize | 4 | 0 | 'Progressive with seizing' | ProgressAndSeize | Progressive with seizing |
| CFGDMPredictAndSeize | 5 | 0 | 'Predictive with seizing' | PredictAndSeize | Predictive with seizing |
| CFGDMPower | 6 | 0 | 'Power' | Power | |
| CFGDMPowerAndSeize | 7 | 0 | 'Power with seizing' | PowerAndSeize | |
| CFGDMPushPreview | 8 | 0 | 'Push Preview' | PushPreview | |
| CFGDMProgressGVP | 9 | 0 | 'Progress GVP' | ProgressGVP | |
| CFGDMPredictGVP | 10 | 0 | 'Predict GVP' | PredictGVP | |
| CFGDMPowerGVP | 11 | 0 | 'Power GVP' | PowerGVP | |

Comments

DM in value name means belonging to CfgDialMode enumeration.

The CfgDialMode is applicable for Configuration Library/Server release 5.1.5xx only.

CfgEnumeratorObjectType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|--------------------------|----|------|-------------------------------|----------|------------------|
| CFGUnknownEnumeratorType | 0 | 0 | 'Unknown Type' | Unknown | |
| | 1 | 0 | 'Media Type' | | |
| | 2 | 0 | 'Service Type' | | |
| | 3 | 0 | 'Customer Segment' | | |
| | 4 | 0 | 'IVR Text To Speech Used' | | |
| | 5 | 0 | 'IVR Speech Recognition Used' | | |
| | 6 | 0 | 'IVR Application Name' | | |
| | 7 | 0 | 'IVR Technical Result' | | |
| | 8 | 0 | 'IVR Technical Result Reason' | | |
| | 9 | 0 | 'Case ID' | | |
| | 10 | 0 | 'Business Result' | | |
| | 11 | 0 | 'Root Interaction ID' | | |

CfgEnumeratorType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|---------------------------------------|----|------|-------------------------------------|---------------------------------|------------------|
| CFGENTNoEnumeratorType | 0 | 0 | 'Unknown' | Unknown | |
| CFGENTInteractionOperationalAttribute | 1 | 0 | 'Interaction Operational Attribute' | InteractionOperationalAttribute | |
| CFGENTRole | 2 | 0 | 'Role' | Role | |
| CFGENTContactAttribute | 3 | 0 | 'Contact Attribute' | ContactAttribute | |
| CFGENTCustom | 4 | 0 | 'Custom' | Custom | |
| CFGENTGVPMasterList | 5 | 0 | 'GVP Master List' | GVPMasterList | |
| CFGENTGVPCustomList | 6 | 0 | 'GVP Custom List' | GVPCustomList | |
| CFGENTGVPMasterDefault | 7 | 0 | 'GVP Master Default' | GVPMasterDefault | |
| CFGENTGVPCustomDefault | 8 | 0 | 'GVP Custom Default' | GVPCustomDefault | |
| CFGENTGVPAlias | 9 | 0 | 'GVP Alias' | GVPAlias | |

Comments

ENT in every enumeration value stands to stress the fact it belongs to CfgEnumeratorType enumeration.

CfgErrorType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|--------------------|----|------|--|-------------------|
| CFGNoError | 0 | 0 | 'Success' | CFGNoError |
| CFGServerNotFound | 1 | 0 | 'SCEE018 - The specified Configuration Server cannot be found. Check the Host name and Port under Details and try again.' | CFGServerNotFound |
| CFGWrongProtocol | 2 | 0 | 'SCEE001 - Your application cannot connect to Configuration Server because it is using a wrong communication protocol. Check the Release Notes for version compatibility information.' | CFGWrongProtocol |
| CFGUnknownMessage3 | | 0 | 'SCEE002 - The application message cannot be identified as a valid message by the Configuration Server. Check the Release Notes for version compatibility information.' | CFGUnknownMessage |
| CFGLoginIncorrect | 4 | 0 | 'SCEE003 - Invalid logon information specified. Check your User name and Password and try again.' | CFGLoginIncorrect |
| CFGNoDBAccess | 5 | 0 | 'SCEE005 - The link to the Configuration Database is not | CFGNoDBAccess |

| Mnemonic | ID | Flag | Short Description | Log Name |
|-------------------------|----|------|---|-------------------------|
| | | | available. Contact your network administrator.' | |
| CFGDBError | 6 | 0 | 'SCEE006 - A database error has occurred. Try the same transaction again. If the problem persists, contact your network administrator.' | CFGDBError |
| CFGAppSecurityViolation | | 0 | 'SCEE004 - Your application is not recognized by the Configuration Server. Check the Application name under Details and try again.' | CFGAppSecurityViolation |
| CFGAccessDenied | 8 | 0 | 'SCEE007 - Access denied. Please contact Technical Support.' | CFGAccessDenied |
| CFGNoPermission | 9 | 0 | 'SCEE008 - Permission error. Your access privileges are insufficient for performing the requested transaction.' | CFGNoPermission |
| CFGObjectNotFound | 10 | 0 | 'SCEE009 - Object does not exist. Another client may have modified or deleted the specified object.' | CFGObjectNotFound |
| CFGListItemNotFound | 11 | 0 | 'SCEE010 - Object association does not exist. Another client may have modified or deleted the specified object association.' | CFGListItemNotFound |
| CFGUniquenessViolation | 12 | 0 | 'SCEE011 - Duplicate information specified. The specified | CFGUniquenessViolation |

| Mnemonic | ID | Flag | Short Description | Log Name |
|------------------------------|----|------|--|----------------------------|
| | | | information is duplicated within the current scope.' | |
| CFGDependentObject13 | 13 | 0 | 'SCEE012 - Additional action required. An additional action is necessary to complete the requested transaction.' | CFGDependentObjects |
| CFGIncompleteObjectData | 14 | 0 | 'SCEE013 - Incomplete data or the specified information is not within the allowable range.' | CFGIncompleteObjectData |
| CFGIncorrectReference15 | 15 | 0 | 'SCEE014 - Your transaction conflicts with existing data. Check the data integrity rules for the object property specified in this error message.' | CFGIncorrectReferences |
| CFGUnchangeableInfo16 | 16 | 0 | 'SCEE015 - Invalid update attempt. The specified data item may not be changed or deleted.' | CFGUnchangeableInfo |
| CFGValueOutOfRange17 | 17 | 0 | 'SCEE016 - Incomplete data or the specified information is not within the allowable range.' | CFGValueOutOfRange |
| CFGUnspecifiedModification18 | 18 | 0 | 'SCEE017 - Unspecified modification. Another user may have just performed the identical transaction.' | CFGUnspecifiedModification |
| CFGDBResponseTimedOut19 | 19 | 0 | 'SCEE019 - The database query timeout expired. Try the transaction again. If the | CFGDBResponseTimedOut |

| Mnemonic | ID | Flag | Short Description | Log Name |
|------------------------|----|------|--|------------------------|
| | | | problem persists, contact the network administrator.' | |
| CFGWrongAPIParameters | 20 | 0 | 'SCEE020 - Compatibility error. Please contact Genesys Technical Support.' | CFGWrongAPIParameters |
| CFGConnWriteError | 21 | 0 | 'SCEE021 - A communication error has occurred during an attempt to send a response or notification from Configuration Server. Check the network connection and restart the application. If the problem persists, contact the network administrator.' | CFGConnWriteError |
| CFGConnReadError | 22 | 0 | 'SCEE022 - A communication error has occurred during the receipt of a response or notification from Configuration Server. Check the Network connection and restart the application. If the problem persists, contact the network administrator.' | CFGConnReadError |
| CFGConfigFileNotRead | 23 | 0 | 'SCEE023 - Compatibility error. Please contact Genesys Technical Support.' | CFGConfigFileNotRead |
| CFGConfigFileIncorrect | 24 | 0 | 'SCEE024 - Compatibility error. Please contact Genesys Technical Support.' | CFGConfigFileIncorrect |
| CFGObjectPackingError | 25 | 0 | 'SCEE025 - The | CFGObjectPackingError |

| Mnemonic | ID | Flag | Short Description | Log Name |
|-------------------------------------|----|------|--|-------------------------------------|
| | | | object cannot be packed for transmission because of object size limitations. Check the Release Notes for version compatibility information.' | |
| CFGUndefinedError | 26 | 0 | 'SCEE026 - Compatibility error. Please contact Genesys Technical Support.' | CFGUndefinedError |
| CFGHistoryLogExpired | 27 | 0 | 'SCEE027 - Compatibility error. Please contact Genesys Technical Support.' | CFGHistoryLogExpired |
| CFGForcedDisconnect | 28 | 0 | 'SCEE028 - The network management system has terminated your communication session. Contact your network administrator.' | CFGForcedDisconnect |
| CFGReadOnlyOperationalModeActivated | 29 | 0 | 'SCEE029 - Transaction rejected. The Configuration Server is currently working in Read-only mode.' | CFGReadOnlyOperationalModeActivated |
| CFGSetOperationalModeFailed | 30 | 0 | 'SCEE030 - Operation failed. The client that activated the Read-only mode is still connected to Configuration Server.' | CFGSetOperationalModeFailed |
| CFGUnsupportedProperty | 31 | 0 | 'SCEE031 - Unsupported object property requested. Please contact Genesys Technical Support.' | CFGUnsupportedProperty |
| CFGUnsupportedObject | 32 | 0 | 'SCEE032 - Unsupported | CFGUnsupportedObject |

| Mnemonic | ID | Flag | Short Description | Log Name |
|--------------------------------|----|------|--|--------------------------------|
| | | | object requested. Please contact Genesys Technical Support.' | |
| CFGExternalAuthenticationError | 35 | 0 | 'SCEE033 - Authentication request is denied.' | CFGExternalAuthenticationError |
| CFGLocaleItemNotChanged | 34 | 0 | 'SCEE034 - Locale element can not be changed. Please contact Genesys Technical Support.' | CFGLocaleItemNotChanged |
| CFGLocaleItemNotFound | 35 | 0 | 'SCEE035 - Locale element can not be located. Please contact Genesys Technical Support.' | CFGLocaleItemNotFound |
| CFGXMLParseError | 36 | 0 | 'SCEE036 - Update data has invalid format. Please contact Genesys Technical Support.' | CFGXMLParseError |
| CFGChangePasswordDenied | 37 | 0 | 'SCEE037 - Password cannot be changed. External authentication is enforced.' | CFGChangePasswordDenied |
| CFGUnsecureConnectionDenied | 38 | 0 | 'SCEE038 - Configuration Server does not accept unsecured connection.' | CFGUnsecureConnectionDenied |
| CFGEmptyPasswordDenied | 39 | 0 | 'SCEE038 - Password cannot be empty.' | CFGEmptyPasswordDenied |

CfgEventType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------|----|------|-------------------|---------------|--|
| CFGError | 0 | 0 | 'Error' | Error | The request has not been processed on the server side. See type CfgErrorCode. |
| CFGRegistered | 1 | 0 | 'Registered' | Registered | The application has been registered to receive unsolicited events regarding objects of a specified type. This event can be expected only upon request ConfRegisterObjectType. |
| CFGUnregistered | 2 | 0 | 'Unregistered' | Unregistered | The application's registration to receive unsolicited events regarding objects of a specified type has been canceled. This event can be expected only upon request ConfUnregisterObjectType. |
| CFGObjectAdded | 3 | 0 | 'Object Added' | ObjectAdded | A new object has been added to the Configuration Database. |
| CFGObjectDeleted | 4 | 0 | 'Object Deleted' | ObjectDeleted | An object has been deleted from the |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------|----|------|-----------------------|-------------------|--|
| | | | | | Configuration Database. |
| CFGObjectInfoChanged | 5 | 0 | 'Object Info Changed' | ObjectInfoChanged | Object information has changed. The parameters of the corresponding delta object structure whose values have not changed are set to zero (NULL). If a parameter of an integer type has changed its current value to zero, the current value is specified in delta object. If a parameter of a character type has changed its current value to an empty string, an empty string is specified in delta object. |
| CFGObjectInfo | 6 | 0 | 'Object Info' | ObjectInfo | Full information about one of the requested objects has been delivered. This event can be expected only upon request ConfGetObjectInfo. |
| CFGObjectCount | 7 | 0 | 'Objects Count' | ObjectCount | Information about the number of objects that satisfy set criteria has been delivered. This event can |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------------|----|------|-----------------------|--------------------|---|
| | | | | | be expected only upon request CongGetObjectCount. |
| CFGEndObjectList | 8 | 0 | 'End Of Objects List' | EndObjectList | Full information or brief information about all requested objects have been transmitted. This event can be expected upon requests ConfGetObjectInfo, ConfGetBriefInfo, and ConfGetACLBriefInfo. |
| CFGDBDisconnect | 9 | 0 | 'DB Disconnected' | DBDisconnected | Link to the database has failed. |
| CFGDBConnected | 10 | 0 | 'DB Connected' | DBConnected | Link to the database has been restored. |
| CFGServerDisconnected | 11 | 0 | 'Server Disconnected' | ServerDisconnected | Communication session with the Configuration Server has failed. |
| CFGClientRegistered | 12 | 0 | 'Client Registered' | ClientRegistered | The requested communication session with the Configuration Server has been established. This event can be expected only upon successful connection to the Configuration Server. |
| CFGUnknownEvent | 13 | 0 | 'Unknown Event' | UnknownEvent | A message received from the Configuration Server cannot |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------|----|------|-------------------|----------------|---|
| | | | | | be recognized. |
| CFGAccessInfo | 14 | 0 | 'Access Info' | AccessInfo | The requested access control list has been delivered. This event can be expected only upon request ConfGetACL. |
| CFGAccessChanged | 15 | 0 | 'Access Changed' | AccessChanged | Access control list of an object has been changed. This event can be expected only upon request ConfSetACL. |
| CFGBriefInfo | 16 | 0 | 'Brief Info' | BriefInfo | Brief information about one of the requested objects has been delivered. This event can be expected only upon request ConfGetBriefInfo. |
| CFGAccountInfo | 17 | 0 | 'Account Info' | AccountInfo | The requested account information has been delivered. This event can be expected only upon request ConfGetAccount. |
| CFGAccountChanged | 18 | 0 | 'Account Changed' | AccountChanged | Account of an application has been changed. This event can be expected only upon request ConfSetAccount. |
| CFGACLBriefInfo | 19 | 0 | 'ACL Brief Info' | ACLBriefInfo | Brief information about a member of an access control list has been delivered. This |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------------|----|------|--------------------------|---------------------|--|
| | | | | | event can be expected only upon request ConfGetACLBriefInfo. |
| CFGEndHistoryLog20 | | 0 | 'End HistoryLog' | EndHistoryLog | All logged notifications have been transmitted. This event can be expected upon requests ConfRestoreSession and ConfGetHistoryLog. |
| CFGOperationalModeSet | | 0 | 'Operational Mode Set' | OperationalModeSet | The operation mode of Configuration Server has been set according to the client request. |
| CFGOperationalMode | | 0 | 'Operational Mode' | OperationalMode | The current operation mode of Configuration Server. This event can be expected only upon request ConfGetOperationalMode. |
| CFGObjectAddedToFolder | | 0 | 'Object Added To Folder' | ObjectAddedToFolder | Obsolete event. Configuration Server no longer sends it. |
| CFGObjectPermissions | | 0 | 'Object Permissions' | ObjectPermissions | Requested access rights on a specific object have been delivered. This event can be expected only upon request ConfGetObjectPermissions. |
| CFGExtObjectInfo 25 | | 0 | 'Ext Object Info' | ExtObjectInfo | Obsolete event. Configuration Server no longer sends it. |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|---------------------------|----|------|----------------------------|------------------------|---|
| CFGReadConfigRegistered | 26 | 0 | 'Read Config Registered' | ReadConfigRegistered | The application has been registered to receive unsolicited events regarding global configuration changes. This event can be expected only upon request ConfRegisterReadConfig. |
| CFGReadConfigUnregistered | 27 | 0 | 'Read Config Unregistered' | ReadConfigUnregistered | The application's registration to receive unsolicited events regarding global configuration changes has been canceled. This event can be expected only upon request ConfUnregisterReadConfig. |
| CFGReadConfig | 28 | 0 | 'Read Config' | ReadConfig | Notification about global configuration change received. Application is supposed to reread its configuration upon receiving of this notification. Starting with 7.0 release, Configuration Server discontinued to send such notifications as a result of change in the permissions processing scheme. |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------|----|------|-------------------|--------------|--|
| CFGSchemaInfo | 29 | 0 | 'Schema Info' | SchemaInfo | Server schema information has been delivered. This event is processed within the library registration process and client code is not supposed to receive it. |
| CFGFilterResult | 30 | 0 | 'Filter Result' | FilterResult | Obsolete event. Configuration Server no longer sends it. |
| CFGLocaleInfo | 31 | 0 | 'Locale Info' | LocaleInfo | Locale information has been received from the Configuration Server. This event can be expected only upon request ConfGetLocaleInfo. |

Comments

For the differences between the content of solicited events (responses) and unsolicited ones (notifications), see CfgEvent.

CfgFieldType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------------|----|------|----------------------|------------------|--|
| CFGFTNoFieldType0 | | 0 | 'Unknown Field Type' | NoFieldType | The value of corresponding parameter has not changed |
| CFGFTRecordID | 1 | 0 | 'Record ID' | RecordID | Record Identifier |
| CFGFTPhone | 2 | 0 | 'Contact Info' | Phone | Phone |
| CFGFTRecordType3 | | 0 | 'Record Type' | RecordType | Record type |
| CFGFTRecordStatus4 | | 0 | 'Record Status' | RecordStatus | Record Status |
| CFGFTDialResult | 5 | 0 | 'Dialing Result' | DialResult | Dial Result |
| CFGFTNumberOfAttempts | | 0 | 'Number of Attempts' | NumberOfAttempts | Number of Attempts |
| CFGFTScheduledTime | | 0 | 'Scheduled Time' | ScheduledTime | Scheduled Time |
| CFGFTCallTime | 8 | 0 | 'Call Time' | CallTime | Call Time |
| CFGFTFrom | 9 | 0 | 'From' | From | Start Time |
| CFGFTUntil | 10 | 0 | 'To' | Until | End Time |
| CFGFTTimeZone | 11 | 0 | 'Time Zone' | TimeZone | Time Zone |
| CFGFTCampaignID12 | | 0 | 'Campaign' | CampaignID | |
| CFGFTAgentID | 13 | 0 | 'Agent' | AgentID | Agent Identifier |
| CFGFTChainID | 14 | 0 | 'Chain' | ChainID | Chain Identifier |
| CFGFTNumberInChain15 | | 0 | 'Number In Chain' | NumberInChain | Number in Chain |
| CFGFTCustomField16 | | 0 | 'User-Defined Field' | CustomField | Customer field. Specified by customer |
| CFGFTANI | 17 | 0 | 'ANI' | ANI | Automatic Number Identification |
| CFGFTLATA | 18 | 0 | 'LATA' | LATA | Local Access and Transport Area |
| CFGFTNPA | 19 | 0 | 'NPA' | NPA | Numbering Plan Area |
| CFGFTNPA_NXX | 20 | 0 | 'NPA-NXX' | NPA-NXX | Numbering |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------|----|------|----------------------|-----------------|---|
| | | | | | plan Area with identifier of specific telephone company central office which serves that number |
| CFGFTStateCode | 21 | 0 | 'State Code' | StateCode | State Code |
| CFGFTInfoDigits | 22 | 0 | 'Info Digits' | InfoDigits | Info digits |
| CFGFTCountryCode | 23 | 0 | 'Country Code' | CountryCode | Country Code |
| CFGFTPhoneType | 24 | 0 | 'Contact Info Type' | PhoneType | Type of phone set |
| CFGFTGroupDBID | 25 | 0 | 'Group' | GroupDBID | AgentGroup or PlaceGroup unique identifier |
| CFGFTAppDBID | 26 | 0 | 'Application' | AppDBID | Application unique identifier |
| CFGFTTreatments | 27 | 0 | 'Treatments History' | Treatments | Treatments history |
| CFGFTMediaReference | 28 | 0 | 'Media Reference' | MediaReference | Reference to media body to be sent in case of treatment |
| CFGFTEmailSubject | 29 | 0 | 'E-mail Subject' | EmailSubject | Email Subject |
| CFGFTEmailTemplateID | 30 | 0 | 'E-mail Template ID' | EmailTemplateID | Email Template ID |
| CFGFTSwitchID | 31 | 0 | 'Switch ID' | SwitchID | Switch ID |

Comments

The CfgFieldType is applicable for Configuration Library/Server release 5.1.5xx only.

FT in value name means belonging to CfgFieldType enumeration.

CfgFilterType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|---------------------------|----|------|-------------------------|--------------------|
| FILTER_NOFILTER | 0 | 0 | 'No Filter' | none |
| FILTER_ACCOUNT_DBID | | 0 | 'Account' | account_dbid |
| FILTER_ACCOUNT_TYPE | | 0 | 'Account Type' | account_type |
| FILTER_CODE_TYPE | 3 | 0 | 'Code Type' | code_type |
| FILTER_APP_PROTOTYPE_DBID | | 0 | 'Application Prototype' | app_prototype_dbid |
| FILTER_APP_DBID | 5 | 0 | 'Application' | app_dbid |
| FILTER_APP_TYPE | 6 | 0 | 'Application Type' | app_type |
| FILTER_ASSOCIATION | 7 | 0 | 'Association' | association |
| FILTER_BACKUP_SERVER_DBID | | 0 | 'Backup Server' | backup_server_dbid |
| FILTER_BRIEF | 9 | 0 | 'Brief' | brief |
| FILTER_CALL_ACTION_CODE | | 0 | 'Call Action Code' | call_action_code |
| FILTER_CALLING_LIST_DBID | | 0 | 'Calling List' | calling_list_dbid |
| FILTER_CALLRESULT_TYPE | | 0 | 'CallResult Type' | call_result |
| FILTER_DBACCESS_DBID | | 0 | 'Database Access Point' | dbaccess_dbid |
| FILTER_DBID | 14 | 0 | 'DBID' | dbid |
| FILTER_DBTABLE_NAME | 15 | 0 | 'Database Table Name' | db_table_name |
| FILTER_DEFAULT_FOLDER | | 0 | 'Default Folder' | default_folder |
| FILTER_DEST_DN_DBID | 17 | 0 | 'Destination DN' | dest_dn_dbid |
| FILTER_DN_DBID | 18 | 0 | 'DN' | dn_dbid |
| FILTER_DN_GROUP_TYPE | | 0 | 'DN Group Type' | dn_group_type |
| FILTER_DN_NUMBER | 20 | 0 | 'DN Number' | dn_number |
| FILTER_DN_TYPE | 21 | 0 | 'DN Type' | dn_type |
| FILTER_EMPLOYEE_ID | 22 | 0 | 'EmployeeID' | employee_id |
| FILTER_FIELD_DBID | 23 | 0 | 'Field' | field_dbid |
| FILTER_FIELD_TYPE | 24 | 0 | 'Field Type' | field_type |
| FILTER_FILTER_DBID | 25 | 0 | 'Filter' | filter_dbid |
| FILTER_FOLDER_DBID | 26 | 0 | 'Folder' | folder_dbid |
| FILTER_FORMAT_DBID | 27 | 0 | 'Format' | format_dbid |

| Mnemonic | ID | Flag | Short Description | Log Name |
|------------------------------|----|------|---------------------------|-----------------------|
| FILTER_GROUP_DBID | 28 | 0 | 'Group' | group_dbid |
| FILTER_HOST_DBID | 29 | 0 | 'Host' | host_dbid |
| FILTER_HOST_TYPE | 30 | 0 | 'Host Type' | host_type |
| FILTER_IS_AGENT | 31 | 0 | 'Is Agent' | is_agent |
| FILTER_APP_IS_SERVER | 32 | 0 | 'Is Server' | is_server |
| FILTER_IVR_DBID | 33 | 0 | 'IVR' | ivr_dbid |
| FILTER_IVR_SERVER_DBID | 34 | 0 | 'IVR Server' | ivr_server_dbid |
| FILTER_LOGIN_DBID | 35 | 0 | 'Login' | login_dbid |
| FILTER_LOG_TABLE_ACCESS_DBID | 36 | 0 | 'Log Table Access' | log_table_access_dbid |
| FILTER_NAME | 37 | 0 | 'Name' | name |
| FILTER_NAME_MSEXPLORER | 38 | 0 | 'MSExplorer Name' | name_msexplorer |
| FILTER_NAME_NETSCAPE | 39 | 0 | 'Netscape Name' | name_netscape |
| FILTER_NO_DBID | 40 | 0 | 'No DBID' | no_dn_dbid |
| FILTER_NO_LOGIN_DBID | 41 | 0 | 'No Login' | no_login_dbid |
| FILTER_NO_PERSON_DBID | 42 | 0 | 'No Person' | no_person_dbid |
| FILTER_NO_PLACE_DBID | 43 | 0 | 'No Place' | no_place_dbid |
| FILTER_OBJECT_TYPE | 44 | 0 | 'Object Type' | object_type |
| FILTER_OFFSET | 45 | 0 | 'Offset' | offset |
| FILTER_OS_TYPE | 46 | 0 | 'OS Type' | os_type |
| FILTER_OWNER_DBID | 47 | 0 | 'Owner' | owner_dbid |
| FILTER_OWNER_TYPE | 48 | 0 | 'Owner Type' | owner_type |
| FILTER_PERSON_DBID | 49 | 0 | 'Person' | person_dbid |
| FILTER_PLACE_DBID | 50 | 0 | 'Place' | place_dbid |
| FILTER_PORT | 51 | 0 | 'Port' | port |
| FILTER_PORT_DBID | 52 | 0 | 'IVR Port' | port_dbid |
| FILTER_PRIMARY_SERVER_DBID | 53 | 0 | 'Primary Server' | primary_server_dbid |
| FILTER_REC_ACTION_CODE | 54 | 0 | 'RecActionCode' | rec_action_code |
| FILTER_SAME_HOST_AND_PORT | 55 | 0 | 'Same Host And Port' | same_host_and_port |
| FILTER_SCS_DBID | 56 | 0 | 'Solution Control Server' | scs_dbid |
| FILTER_SCRIPT_DBID | 57 | 0 | 'Script' | script_dbid |
| FILTER_SCRIPT_TYPE | 58 | 0 | 'Script Type' | script_type |
| FILTER_SERVER_DBID | 59 | 0 | 'Server' | server_dbid |
| FILTER_SKILL_DBID | 60 | 0 | 'Skill' | skill_dbid |
| FILTER_STATE | 61 | 0 | 'State' | state |
| FILTER_SWITCH_DBID | 62 | 0 | 'Switch' | switch_dbid |

| Mnemonic | ID | Flag | Short Description | Log Name |
|----------------------------------|----|------|---------------------------------|---------------------------|
| FILTER_TABLE_ACCESS_DBID | 63 | 0 | 'Table Access' | table_access_dbid |
| FILTER_TABLE_DBID | 64 | 0 | 'Table' | table_dbid |
| FILTER_TABLE_TYPE | 65 | 0 | 'Table Type' | table_type |
| FILTER_TENANT_DBID | 66 | 0 | 'Tenant' | tenant_dbid |
| FILTER_TREATMENT_DBID | 67 | 0 | 'Treatment' | treatment_dbid |
| FILTER_TSERVER_DBID | 68 | 0 | 'TServer' | tserver_dbid |
| FILTER_TYPE | 69 | 0 | 'Type' | type |
| FILTER_USER_NAME | 70 | 0 | 'User Name' | user_name |
| FILTER_TABLE_ACCESS_TYPE | 71 | 0 | 'TableAccess Type' | type |
| FILTER_LOGIN_CODE | 72 | 0 | 'Login Code' | login_code |
| FILTER_DN_ALIAS | 73 | 0 | 'DN Alias' | dn_alias |
| FILTER_DATA | 74 | 0 | 'Data' | data |
| FILTER_ALL_TENANTS | 75 | 0 | 'All Tenants' | all_tenants |
| FILTER_APP_VERSION | 76 | 0 | 'Application Version' | version |
| FILTER_NO_SWITCH_DBID | 77 | 0 | 'No Switch' | no_switch_dbid |
| FILTER_OBJECT_DBID | 78 | 0 | 'Object' | object_dbid |
| FILTER_IS_PRIMARY | 79 | 0 | 'Is Primary' | is_primary |
| FILTER_MAXVALUE | 80 | 0 | 'MaxValue' | none |
| FILTER_IVR_PORT_NUMBER | 81 | 0 | 'IVR Port Number' | port_number |
| FILTER_ALLOWED_DNS | 82 | 0 | 'Allowed DNSs' | allowed_dns |
| FILTER_NO_CLIENT_DBID | 83 | 0 | 'No client' | no_client_dbid |
| FILTER_FIRST_NAME | 84 | 0 | 'First name' | first_name |
| FILTER_LAST_NAME | 85 | 0 | 'Last name' | last_name |
| FILTER_CAPACITY_TENANT_DBID | 86 | 0 | 'Tenant for Capacity Rule' | capacity_tenant_dbid |
| FILTER_CAPACITY_AGENT_DBID | 87 | 0 | 'Agent for Capacity Rule' | capacity_agent_dbid |
| FILTER_CAPACITY_PLACE_DBID | 88 | 0 | 'Place for Capacity Rule' | capacity_place_dbid |
| FILTER_CAPACITY_AGENT_GROUP_DBID | 89 | 0 | 'Agent Group for Capacity Rule' | capacity_agent_group_dbid |
| FILTER_CAPACITY_PLACE_GROUP_DBID | 90 | 0 | 'Place Group for Capacity Rule' | capacity_place_group_dbid |
| FILTER_DELEGATE_DBID | 91 | 0 | 'Delegate account' | delegate_dbid |
| FILTER_DELEGATE_TYPE | 92 | 0 | 'Delegate account type' | delegate_type |
| FILTER_ENUMERATOR_TYPE | 93 | 0 | 'Business Attribute type' | enumerator_type |

| Mnemonic | ID | Flag | Short Description | Log Name |
|---------------------------|-----|------|----------------------------|------------------|
| FILTER_ENUMERATOR_DBID | 94 | 0 | 'Business Attribute' | enumerator_dbid |
| FILTER_DEFAULT_VALUE | 95 | 0 | 'Default value' | default_value |
| FILTER_EXCLUDE_BYTECODE | 96 | 0 | 'Exclude bytecode?' | exclude_bytecode |
| FILTER_REGISTER_ALL | 97 | 0 | 'Register all flag' | register_all |
| FILTER_OBJECT_PATH | 98 | 0 | 'Object path needed' | object_path |
| FILTER_DISPLAY_NAME | 99 | 0 | 'Display name' | display_name |
| FILTER_STAT_DAY_DBID | 100 | 0 | 'Statistical Day' | stat_day_dbid |
| FILTER_CMP_INSENSITIVE | 101 | 0 | 'Compare Insensitive flag' | cmp_insensitive |
| FILTER_READ_FOLDER_DBID | 102 | 0 | 'Read folder flag' | read_folder_dbid |
| FILTER_CAMPAIN_DBID | 103 | 0 | 'Campaign' | campaign_dbid |
| FILTER_RESELLER_DBID | 104 | 0 | 'Reseller' | reseller_dbid |
| FILTER_CUSTOMER_DBID | 105 | 0 | 'Customer' | customer_dbid |
| FILTER_FOLDER_CLASS | 106 | 0 | 'Folder Class' | folder_class |
| FILTER_OBJECTIVETABLETYPE | 107 | 0 | 'Objective Table Type' | objtable_type |
| FILTER_STATDAY_TYPE | 108 | 0 | 'Stat Day Type' | statday_type |
| FILTER_TASK_TYPE | 109 | 0 | 'Task Type' | task_type |

CfgFlag

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------|-----------|-------------|--------------------------|-----------------|---|
| CFGNoFlag | 0 | 0 | 'Unknown' | NoFlag | The value of the corresponding parameter has not changed. |
| CFGFalse | 1 | 0 | 'False' | False | Negative condition. |
| CFGTrue | 2 | 0 | 'True' | True | Positive condition. |

CfgFolderClass

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|-----------------------------|----|------|--------------------------|------------------------|
| CFGFCNoClass | 0 | 0 | 'Unknown' | Unknown |
| CFGFCDefault | 1 | 0 | 'Default' | Default |
| CFGFCSite | 2 | 0 | 'Site' | Site |
| CFGFCGVPCustomerGroup | 3 | 0 | 'GVP Customer Group' | GVPCustomerGroup |
| CFGFCGVPIVRProfileGroup | 4 | 0 | 'GVP IVRProfile Group' | GVPIVRProfileGroup |
| CFGFCGVPRoot | 5 | 0 | 'GVP Root' | GVPRoot |
| CFGFCHostGroup | 6 | 0 | 'Host Group' | HostGroup |
| CFGFCGVPCustomerGroupRoot | 7 | 0 | 'GVP Customer Groups' | GVPCustomerGroupRoot |
| CFGFCGVPIVRProfileGroupRoot | 8 | 0 | 'GVP IVR Profile Groups' | GVPIVRProfileGroupRoot |
| CFGFCHostGroupRoot | 9 | 0 | 'Host Groups' | HostGroupRoot |
| CFGFCGVPDIDGroups | 10 | 0 | 'DID Groups' | GVPDIDGroups |
| CFGFCGVPDIDs | 11 | 0 | 'DIDs' | GVPDIDs |

CfgGroupType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|--------------------|----|------|----------------------|------------|---|
| CFGNoGroup | 0 | 0 | 'Unknown Group Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGAgentGroupType | 1 | 0 | 'Agent Group' | AgentGroup | |
| CFGPlaceGroupType | 2 | 0 | 'Place Group' | PlaceGroup | |
| CFGDNGroupType | 3 | 0 | 'DN Group' | DNGroup | |
| CFGAccessGroupType | 4 | 0 | | | |

Comments

This data type is not used anywhere in the API functions or events.

CfgHAType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------|----|------|-------------------|-------------|--|
| CFGHTNoHAType | 0 | 0 | 'Unknown HA Type' | Unknown | The value of corresponding parameter has not changed |
| CFGHTColdStandby | 1 | 0 | 'Not Specified' | ColdStandby | A cold standby |
| CFGHTWarmStandby | 2 | 0 | 'Warm Standby' | WarmStandby | A warm standby |
| CFGHTHotStandby | 3 | 0 | 'Hot Standby' | HotStandby | A hot standby |

Comments

The CfgHAType specifies the types of high availability that will be possible to use by Genesys or third party application. The behavior of the application/client depends upon high availability type specified in server this application is connected to.

HT in every enumeration value stands to stress the fact it belongs to CfgHAType enumeration.

CfgHostType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------------|----|------|-----------------------|--------------------|---|
| CFGNoHost | 0 | 0 | 'Unknown Host Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGNetworkServer1 | | 0 | 'Network Server' | NetworkServer | A major computer on the Service Provider data network. |
| CFGTenantServer 2 | | 0 | 'Tenant Server' | TenantServer | A major computer on a tenant data network. |
| CFGServiceWorkstation | | 0 | 'Service Workstation' | ServiceWorkstation | A workstation on the Service Provider data network which is not an agent workstation. |
| CFGTenantWorkstation | | 0 | 'Tenant Workstation' | TenantWorkstation | A workstation on a tenant data network which is not an agent workstation. |
| CFGAgentWorkstation | | 0 | 'Agent Workstation' | AgentWorkstation | An agent workstation. |
| CGFAuxComputer 6 | | 0 | 'Auxiliary Computer' | AuxComputer | An auxiliary computer on either the Service Provider network or a tenant network. |
| CFGGVPWorkstation | | 0 | 'GVP Workstation' | GVPWorkstation | |

CfgIVRProfileType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|-----------------|-----------|-------------|--------------------------|-----------------|
| CFGIPType | 0 | 0 | 'Unknown' | Unknown |
| CFGIPMaxType | 1 | 0 | 'Inbound' | CFGIPMaxType |
| CFGIPOutbound | 2 | 0 | 'Outbound' | Outbound |

CfgIVRType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------------------|----|------|---------------------------|----------------------|--|
| CFGIVRTNoIVRType0 | | 0 | 'Unknown IVR Type' | Unknown | The value of corresponding parameter has not changed |
| CFGIVRTConversant1 | | 0 | 'Conversant' | Conversant | |
| CFGIVRTDirectTalk6000 | | 0 | 'WVR for AIX' | DirectTalk6000 | |
| CFGIVRTSyntellectVocalPoint | | 1 | 'Syntellect Vocal Point' | SyntellectVocalPoint | |
| CFGIVRTSyntellectPremier | | 1 | 'Syntellect Premier' | SyntellectPremier | |
| CFGIVRTSyntellectVista | | 1 | 'Syntellect Vista' | SyntellectVista | |
| CFGIVRTVoiceTek 6 | | 1 | 'Voicetek' | VoiceTek | |
| CFGIVRTAgility 7 | | 1 | 'Agility' | Agility | |
| CFGIVRTMeridianIntegrated | | 1 | 'Meridian Integrated' | MeridianIntegrated | |
| CFGIVRTSymposiumOpen | | 1 | 'Symposium Open' | SymposiumOpen | |
| CFGIVRTEdify 10 | | 0 | 'Edify' | Edify | |
| CFGIVRTBrite 11 | | 1 | 'Brite' | Brite | |
| CFGIVRTShowNTel12 | | 0 | 'ShowNTel' | ShowNTel | |
| CFGIVRTIntervoice13 | | 0 | 'Intervoice Brite' | Intervoice | |
| CFGIVRTPeriphonics14 | | 0 | 'Periphonics VPS/is' | Periphonics | |
| CFGIVRTAmerex 15 | | 1 | 'Amerex' | Amerex | |
| CFGIVRTDirectTalkNT | | 0 | 'WVR for Windows' | DirectTalkNT | |
| CFGIVRTeleraGVP 17 | | 0 | 'Genesys Voice Platform' | TeleraGVP | |
| CFGIVRTMPS 18 | | 0 | 'MPS' | MPS | |
| CFGIVRTAspectCSS19 | | 0 | 'Aspect CSS' | AspectCSS | |
| CFGIVRTMSSpeechServer | | 0 | 'Microsoft Speech Server' | MSSpeechServer | |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|---------------------|-----------|-------------|--------------------------|-----------------|-------------------------|
| CFGIVRTOtherIVRType | 21 | 0 | 'Other IVR Type' | OtherIVRType | |
| CFGIVRTEnvoy | 22 | 0 | 'Envoy' | Envoy | |

Comments

IVRT in value name means belonging to CfgIVRType enumeration.

CfgLanguage

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------|----|------|------------------------|----------------|------------------|
| CFGLNGNoLanguage | 0 | 0 | 'Unknown Language' | Unknown | |
| CFGLNGEnglish | 1 | 0 | 'English' | English | |
| CFGLNGFrench | 2 | 0 | 'French' | French | |
| CFGLNGGerman | 3 | 0 | 'German' | German | |
| CFGLNGItalian | 4 | 0 | 'Italian' | Italian | |
| CFGLNGLASpanish | 5 | 0 | 'Spanish' | LASpanish | |
| CFGLNGBrasPortuguese | 6 | 0 | 'Brazilian Portuguese' | BrasPortuguese | |
| CFGLNGJapanese | 7 | 0 | 'Japanese' | Japanese | |
| CFGLNGKorean | 8 | 0 | 'Korean' | Korean | |
| CFGLNGTradChinese | 9 | 0 | 'Traditional Chinese' | TradChinese | |
| CFGLNGSimplChinese | 10 | 0 | 'Simplified Chinese' | SimpleChinese | |

Comments

LNG in every enumeration value stands to stress the fact it belongs to CfgLanguage enumeration.

The CfgLanguage specifies the language that is used for localization.

CfgLinkType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------------|----|------|------------------------|--------------------|---|
| CFGNoLink | 0 | 0 | 'Unknown Link Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGMeridianLink41 | | 0 | 'Meridian Link 4' | MeridianLink4 | Corresponds to switch type CFGNortelMeridian. |
| CFGMeridianLink52 | | 0 | 'Meridian Link 5' | MeridianLink5 | Corresponds to switch type CFGNortelMeridian. |
| CFGGalaxyLink | 3 | 0 | 'Galaxy Link' | GalaxyLink | Corresponds to switch type CFGRockwellGalaxy. |
| CFGspectrumLink4 | | 0 | 'Spectrum Link' | SpectrumLink | Corresponds to switch type CFGRockwellSpectrum. |
| CFGCompuCall055 | 5 | 0 | 'SCAI 7' | CompuCall05 | Corresponds to switch type CFGNortelDMS100. |
| CFGCompuCall066 | 6 | 0 | 'SCAI 8' | CompuCall06 | Corresponds to switch type CFGNortelDMS100. |
| CFGCallVisorASAI7 | 7 | 0 | 'Call Visor ASAI' | CallVisorASAI | Corresponds to switch type CFGLucentDefinityG3. |
| CFGEthernetASAI8 | 8 | 0 | 'Ethernet ASAI' | EthernetASAI | Corresponds to switch type CFGLucentDefinityG3. |
| CFGApplicationBridge5 | 10 | 0 | 'Application Bridge 5' | ApplicationBridge5 | Corresponds to switch type CFGAspectCallCenter. |
| CFGApplicationBridge6 | 10 | 0 | 'Application Bridge 6' | ApplicationBridge6 | Corresponds to switch type CFGAspectCallCenter. |
| CFGCallBridge2 | 11 | 0 | 'CallBridge 2' | CallBridge2 | Corresponds to switch type CFGRolm9751CBX. |
| CFGCallBridge3 | 12 | 0 | 'CallBridge 3' | CallBridge3 | Corresponds to |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------------------|----|------|---|-------------------------------|---|
| | | | | | switch type CFGRolm9751CBX. |
| CFGOAI | 13 | 0 | 'OAI' | OAI | Corresponds to switch type CFGIntecomIBX80. |
| CFGApplicationLink4 | 14 | 0 | 'Application Link' | ApplicationLink | Corresponds to switch type CFGEricssonMD110. |
| CFGPinnacle | 15 | 0 | 'Pinnacle' | Pinnacle | Corresponds to switch type CFGLucent5ESS. |
| CFGMadgeLink | 16 | 0 | 'Madge Link' | MadgeLink | Corresponds to switch type CFGMadge. |
| CFGNECLink | 17 | 0 | 'NEC Link' | NECLink | Corresponds to switch type CFGNEC. |
| CFGFujitsuLink | 18 | 0 | 'Fujitsu Link' | FujitsuLink | Corresponds to switch type CFGFujitsu |
| CFGHostInterfaceLink | 19 | 0 | 'Host Interface Link' | HostInterfaceLink | Corresponds to switch type CFGHarrisVoiceFrame. |
| CFGWorkStationInterfaceLink | 20 | 0 | 'Workstation Interface Link' | WorkStationInterfaceLink | Corresponds to switch type CFGHarrisVoiceFrame. |
| CFGGateway01Link | 21 | 0 | 'Gateway 01 Link' | Gateway01Link | |
| CFGApplicationConnectivityLink10 | 22 | 0 | 'Application Connectivity Link 1.x' | ApplicationConnectivityLink10 | Corresponds to switch type CFGSiemensHicom150. |
| CFGApplicationConnectivityLink20 | 23 | 0 | 'Application Connectivity Link 2.x' | ApplicationConnectivityLink20 | Corresponds to switch type CFGSiemensHicom150. |
| CFGCallBridgeACLISDN0ISO | 24 | 0 | 'CallBridge ACL ISDN SO ISO' | CallBridge/ACL ISDN SO ISO | Corresponds to switch type CFGSiemensHicom300. |
| CFGCallBridgeACLV24ISO | 25 | 0 | 'CallBridge ACL v.24 ISO' | CallBridge/ACL V24 ISO | Corresponds to switch type CFGSiemensHicom300. |
| CFGiSLinkCSTAI | 26 | 0 | 'iS Link CSTA I' | iSLinkCSTA I | Corresponds to switch type CFGPhilipsSophoS300. |
| CFGMatraLinkCSTAI | 27 | 0 | 'Matra Link CSTA II' | MatraLinkCSTA II | Corresponds to switch types CFGMatraMC6500 and |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------|----|------|----------------------------|------------------------|---|
| | | | | | CFGMatraMC7500. |
| CFGApplicationLink | 28 | 0 | 'Application Link CSTA' | ApplicationLinkCSTA | Corresponds to switch type CFGEricssonACP1000. |
| CFGiCAT23ICCL2 | 29 | 0 | 'iCAT 2.3 ICCL 2' | iCAT2.3(ICCL2) | Corresponds to switch type CFGSiemensGECiSDX. |
| CFGiCAT3xICCL3 | 30 | 0 | 'iCAT 3.x ICCL 3' | iCAT3.x(ICCL3) | Corresponds to switch type CFGSiemensGECiSDX. |
| CFGApplicationLink | 31 | 0 | 'Application Link CSTA I' | ApplicationLinkCSTA I | Corresponds to switch type CFGAlcatelA4400DHS3. |
| CFGApplicationLink | 32 | 0 | 'Application Link CSTA II' | ApplicationLinkCSTA II | Corresponds to switch type CFGAlcatelA4400DHS3. |
| CFGGenericLink | 33 | 0 | 'Generic Link' | GenericLink | Corresponds to switch type CFGGenericSwitch. |
| CFGCompuCall09 | 34 | 0 | 'SCAI 11' | CompuCall09 | Corresponds to switch type CFGNortelDMS100. |
| CFGCompuCall10 | 35 | 0 | 'SCAI 12' | CompuCall10 | Corresponds to switch type CFGNortelDMS100. |
| CFGCompuCall11 | 36 | 0 | 'SCAI 13' | CompuCall11 | Corresponds to switch type CFGNortelDMS100. |
| CFGApplicationBridge | 37 | 0 | 'Application Bridge 7' | ApplicationBridge7 | Corresponds to switch type CFGAspectCallCenter. |
| CFGCallBridge4 | 38 | 0 | 'CallBridge4' | CallBridge4 | Corresponds to switch type CFGRolm9751CBX. |
| CfgDelcoACD | 39 | 0 | 'Delco ACD' | DelcoACD | Corresponds to switch type CFGDelcoACD. |
| CFGHitachiCx8000 | 40 | 0 | 'Hitachi Cx8000' | HitachiCx8000 | Corresponds to switch type CFGHitachiCX8000. |
| CFGStarexLink | 41 | 0 | 'Starex Link' | StarexLink | Corresponds to switch type CFLGStarex. |
| CFGMiTai7_3 | 42 | 0 | 'MiTai 7.3' | MiTai7_3 | Corresponds to switch type CFGMitelSX2000. |
| CFGMeridianLink | 43 | 0 | 'Meridian Link' | MeridianLinkServices | Corresponds to |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------------------------|----|------|--------------------------------------|-----------------------------|--|
| | | | ServicesSymposium' | | switch type CFGMeridianCallCenter |
| CFGApplicationConnectivityLinkCSTA | 44 | 0 | 'Application Connectivity Link CSTA' | ApplicationConnectivityLink | Corresponds to switch type CFGSiemensHicom150E. |
| CFGCallBridgeDX | 45 | 0 | 'CallBridge DX' | CallBridgeDX | Corresponds to switch type CFGSiemensRealtisDX. |
| CFGCoralLink | 46 | 0 | 'Coral Link' | CoralLink | |

CfgMediaType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|-----------------------|------|------|------------------------|--------------------|
| CFGMediaVoice | 100 | 0 | 'Media Voice' | MediaVoice |
| CFGMediaVoIP | 101 | 0 | 'MediaVoIP' | MediaVoIP |
| CFGMediaEMail | 102 | 0 | 'Media Voice' | MediaEMail |
| CFGMediaVMail | 103 | 0 | 'Media EMail' | MediaVMail |
| CFGMediaSMail | 104 | 0 | 'Media VMail' | MediaSMail |
| CFGMediaChat | 105 | 0 | 'Media SMail' | MediaChat |
| CFGMediaVideo | 106 | 0 | 'Media Chat' | MediaVideo |
| CFGMediaCobrowsing | 107 | 0 | 'Media Cobrowsing' | MediaCobrowsing |
| CFGMediaWhiteboard | 108 | 0 | 'Media Whiteboard' | MediaWhiteboard |
| CFGMediaAppSharing | 109 | 0 | 'Media AppSharing' | MediaAppSharing |
| CFGMediaWebForm | 110 | 0 | 'Media WebForm' | MediaWebForm |
| CFGMediaWorkItem | 111 | 0 | 'Media Work Item' | MediaWorkItem |
| CFGMediaCallback | 112 | 0 | 'Media Callback' | MediaCallback |
| CFGMediaFax | 113 | 0 | 'Media Fax' | MediaFax |
| CFGMediaIMChat | 114 | 0 | 'Media IMChat' | MediaIMChat |
| CFGMediaBusinessEvent | 115 | 0 | 'Media Business Event' | MediaBusinessEvent |
| CFGMediaAlert | 116 | 0 | 'Media Alert' | MediaAlert |
| CFGMediaSMS | 117 | 0 | 'Media SMS' | MediaSMS |
| CFGMediaDefault | 118 | 0 | 'Media Any' | MediaAny |
| CFGMediaSMSSession | 119 | | | |
| CFGMediaMMS | 120 | | | |
| CFGMediaMMSSession | 121 | | | |
| CFGMediaCustom0 | 1000 | 0 | 'Media Custom0' | MediaCustom0 |

CfgOSType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------|----|------|------------------------------------|-------------------|---|
| CFGNoOS | 0 | 0 | 'Unknown OS Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGSolaris | 1 | 0 | 'Solaris' | Solaris | |
| CFGSolarisX86 | 2 | 0 | 'Solaris x86' | SolarisX86 | |
| CFGDigitalUNIX | 3 | 0 | 'Tru64 Unix' | DigitalUNIX | |
| CFGHPUX | 4 | 0 | 'HP-UX' | HPUX | |
| CFGaix | 5 | 0 | 'IBM AIX' | AIX | |
| CFGSunOS | 6 | 0 | 'SunOS' | SunOS | |
| CFGWinNT | 7 | 0 | 'Windows NT' | WinNT | |
| CFGWindows | 8 | 0 | 'Windows' | Windows | |
| CFGOS2 | 9 | 0 | 'IBM OS2' | OS/2 | |
| CFGMacintosh | 10 | 0 | 'Macintosh' | UMacintosh | |
| CFGTandemUNIX | 11 | 0 | 'Tandem UNIX' | Tandem UNIX | |
| CFGUnixWare | 12 | 0 | 'UNIX Ware' | Unixware | |
| CFGWindows2000 | 13 | 0 | 'Windows 2000' | Windows2000 | |
| CFGWindowsXP | 14 | 0 | 'Windows XP' | WindowsXP | |
| CFGWindowsServer2003 | 15 | 0 | 'Windows Server 2003' | WindowsServer2003 | |
| CFGRedHatLinux | 16 | 0 | 'RedHat Enterprise Linux AS/Intel' | RedHatLinux | |
| CFGWindowsServer2008 | 17 | | | | |
| CFGWindowsVista | 18 | | | | |
| CFGMaxOSType | 19 | | | | |

CfgObjectState

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------|----|------|------------------------|----------|---|
| CFGNoObjectState0 | | 0 | 'Unknown Object State' | Unknown | The value of the corresponding parameter has not changed. |
| CFGEnabled | 1 | 0 | 'Enabled' | Enabled | Object can be used. Indicates that an entity represented by this object is in a regular operating condition and can be used without any restrictions. |
| CFGDisabled | 2 | 0 | 'Disabled' | Disabled | Object cannot be used. In version 5.1, applies only to the objects that represent call targets in call-processing algorithms and generally means that a call cannot be directed to this target, even if the real-time status information received through the CTI interface indicates that this target is ready to accept a call. |

CfgObjectType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------|----|------|-------------------|----------|---|
| CFGNoObject | 0 | 0 | 'Unknown Object' | Unknown | Object is not defined. |
| CFGSwitch | 1 | 0 | 'Switch' | Switch | An aggregate of telephony objects within one telephone switch controlled through one T-Server. See structure CfgSwitch in section Configuration Data Types. |
| CFGDN | 2 | 0 | 'DN' | DN | A telephony object uniquely identified by its directory number within a switch at which telephone calls may reside and be handled. See structure CfgDN in section Configuration Data Types. |
| CFGPerson | 3 | 0 | 'Person' | Person | A person registered in the Configuration Database. See structure CfgPerson in section Configuration Data Types. |
| CFGPlace | 4 | 0 | 'Place' | Place | A location that has one or more |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------|----|------|-------------------|-------------|---|
| | | | | | telephony objects operated by one agent at a time. See structure CfgPlace in section Configuration Data Types. |
| CFGAgentGroup | 5 | 0 | 'Agent Group' | AgentGroup | A group of agents. See structure CfgAgentGroup in section Configuration Data Types. |
| CFGPlaceGroup | 6 | 0 | 'Place Group' | PlaceGroup | A group of places. See structure CfgPlaceGroup in section Configuration Data Types. |
| CFGTenant | 7 | 0 | 'Tenant' | Tenant | A business subscribed to one or more CTI services. See structure CfgTenant in section Configuration Data Types. |
| CFGService | 8 | 0 | 'Solution' | Service | CTI functionality offered to tenants as a service. See structure CfgService in section Configuration Data Types. |
| CFGApplication | 9 | 0 | 'Application' | Application | An application that provides some of the CTI functions. See structure CfgApplication in section Configuration Data Types. |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|---------------------|----|------|--------------------|----------------|---|
| CFGHost | 10 | 0 | 'Host' | Host | A computer where CTI software is loaded. See structure CfgHost in section Configuration Data Types. |
| CFGPhysicalSwitch11 | | 0 | 'Switching Office' | PhysicalSwitch | A telephone office that contains and/or terminates telephone objects of one or more tenants. See structure CfgPhysicalSwitch in section Configuration Data Types. |
| CFGScript | 12 | 0 | 'Script' | Script | A set of call treatments created within a service. See structure CfgScript in section Configuration Data Types. |
| CFGSkill | 13 | 0 | 'Skill' | Skill | Agent skill. See structure CfgSkill in section Configuration Data Types. |
| CFGActionCode | 14 | 0 | 'Action Code' | ActionCode | Action code. See structure CfgActionCode in section Configuration Data Types. |
| CFGAgentLogin | 15 | 0 | 'Agent Login' | AgentLogin | Agent identifier in a switching system. See structure CfgAgentLogin in section Configuration Data Types. |
| CFGTransaction | 16 | 0 | 'Transaction' | Transaction | Customer- |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------|----|------|------------------------|--------------|---|
| | | | | | defined transaction associated with an object type. See structure CfgTransaction in section Configuration Data Types. |
| CFGDNGroup | 17 | 0 | 'DN Group' | DNGroup | A group of DNs. See structure CfgDNGroup in section Configuration Data Types. |
| CFGStatDay | 18 | 0 | 'Statistical Day' | StatDay | A statistical day. See structure CfgStatDay in section Configuration Data Types. |
| CFGStatTable | 19 | 0 | 'Statistical Table' | StatTable | A statistical table. See structure CfgStatTable in section Configuration Data Types. |
| CFGAppPrototype | 20 | 0 | 'Application Template' | AppPrototype | An application template that contains application-specific options and their default values where appropriate. See structure CfgAppPrototype in section Configuration Data Types. |
| CFGAccessGroup | 21 | 0 | 'Access Group' | AccessGroup | A group of persons that have the same set of permissions with respect to Configuration Data objects. |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------|----|------|-------------------|-------------|---|
| | | | | | See structure CfgAccessGroup in section Access Control Functions and Data Types. |
| CFGFolder | 22 | 0 | 'Folder' | Folder | A logical entity that defines a certain object type under a particular parent object. See structure CfgFolder in section Access Control Functions and Data Types. |
| CFGField | 23 | 0 | 'Field' | Field | A basic subdivision of data record |
| CFGFormat | 24 | 0 | 'Format' | Format | Arrangement of fields within a document/database table |
| CFGTableAccess | 25 | 0 | 'Table Access' | TableAccess | A logical entity that defines document access information |
| CFGCallingList | 26 | 0 | 'Calling List' | CallingList | A calling list. See structure CfgCallingList in section Configuration Data Types. |
| CFGCampaign | 27 | 0 | 'Campaign' | Campaign | Outbound campaign. See structure CfgCampaign in section Configuration Data Types. |
| CFGTreatment | 28 | 0 | 'Treatment' | Treatment | A logical entity that defines further step(s) call processing based on call result. |
| CFGFilter | 29 | 0 | 'Filter' | Filter | A logical entity that allows to |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------|----|------|----------------------|----------------|---|
| | | | | | keep away all the staff that is not required from document the application uses. |
| CFGTimeZone | 30 | 0 | 'Time Zone' | TimeZone | A logical entity that gives the time difference between GMT and local time for both standard time and summertime (Daylight Savings Time). |
| CFGVoicePrompt | 31 | 0 | 'Voice Prompt' | VoicePrompt | A voice prompt. |
| CFGIVRPort | 32 | 0 | 'IVR Port' | IVRPort | A telephony object uniquely identified by its number within an IVR at which telephone calls may reside and be handled. |
| CFGIVR | 33 | 0 | 'IVR' | IVR | An aggregate of IVR ports within one IVR controlled through one IVR interface driver. See structure CfgIVR in section Configuration Data Types. |
| CFGAlarmCondition | 34 | 0 | 'Alarm Condition' | AlarmCondition | An alarm condition |
| CFGEnumerator | 35 | 0 | 'Business Attribute' | Enumerator | An object representing Business Attribute entity which serves as a collection of Attribute Value (CFGEnumeratorValue) objects. See structure |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|--------------------|----|------|----------------------------|------------------|---|
| | | | | | CfgEnumerator in section Configuration Data Types. |
| CFGEnumeratorValue | 36 | 0 | 'Business Attribute Value' | EnumeratorValue | An object representing AttributeValue entity. See structure CfgEnumeratorValue in section Configuration Data Types. |
| CFGObjectiveTable | 37 | 0 | 'Objective Table' | ObjectiveTable | An objective table. See structure CfgObjectiveTable in section Configuration Data Types. |
| CFGCampaignGroup | 38 | 0 | 'Campaign Group' | CampaignGroup | |
| CFGGVPReseller | 39 | 0 | 'GVP Reseller' | GVPReseller | |
| CFGGVPCustomer | 40 | 0 | 'GVP Customer' | GVPCustomer | |
| CFGGVPiVRProfile | 41 | 0 | 'GVP IVRProfile' | GVPIVRProfile | |
| CFGScheduledTask | 42 | 0 | 'Scheduled Task' | ScheduledTask | |
| CFGRole | 43 | | | | |
| CFGPersonLastLog | 44 | | | | |
| CFGMaxObjectType | 45 | 0 | 'Shortcut' | CFGMaxObjectType | A maximum value of this enumeration type. |

CfgObjectiveTableType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|--------------------|-----------|-------------|--------------------------|-----------------|
| CFGOTTNoType | 0 | 0 | 'Unknown' | Unknown |
| CFGOTTDefault | 1 | 0 | 'Default' | Default |
| CFGOTTCostContract | 2 | 0 | 'Cost Contract' | CostContract |

CfgOperationMode

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------|----|------|--------------------------|-----------------|--|
| CFGOMNoOperationMode | | 0 | 'Unknown Operation Mode' | NoOperationMode | The value of corresponding parameter has not changed |
| CFGOMManual | 1 | 0 | 'Manual' | Manual | Manual |
| CFGOMSchedule | 2 | 0 | 'Scheduled' | Schedule | Scheduled |

Comments

TT in value name means belonging to CfgOperationMode enumeration.

The CfgOperationMode is applicable for Configuration Library/Server release 5.1.5xx only.

CfgOperationalMode

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------------|----|------|----------------------|-------------------------|--|
| CFGOperationalNoMode | 0 | | 'Operational NoMode' | OperationalNoMode | |
| CFGOperationalFullMode | 0 | | 'Full Mode' | OperationalFullMode | Normal mode of Configuration Server operation. |
| CFGOperationalReadOnlyMode | 0 | | 'Read-Only' | OperationalReadOnlyMode | No modifications to the Configuration Database are allowed in this mode. |

CfgOptimizationMethod

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|---------------------------|----|------|---------------------------------|------------------------|--|
| CFGOMNoOptimizationMethod | 0 | 0 | 'Unknown Optimization Criteria' | OMNoOptimizationMethod | The value of corresponding parameter has not changed |
| CFGOMBusyFactor1 | 1 | 0 | 'Agent Busy Factor' | OMBusyFactor | A target busy factor. |
| CFGOMOverdialRate | 2 | 0 | 'Overdial Rate' | OMOverdialRate | An over dial rate. |
| CFGOMWaitTime | 3 | 0 | 'Average Waiting Time' | OMWaitTime | A wait time between calls. |

Comments

OM in value name means belonging to CfgOptimizationMethod enumeration.

The CfgOptimizationMethod is applicable for Configuration Library/Server release 5.1.5xx only.

CfgPermissions

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------------|-----------|-------------|--------------------------|-----------------|-------------------------|
| NoAccess | 0 | 0 | | | |
| ReadAccess | 1 | 0 | | | |
| CreateAccess | 2 | 0 | | | |
| ChangeAccess | 4 | 0 | | | |
| ExecuteAccess | 8 | 0 | | | |
| ReadExecuteAccess | 16 | 0 | | | |
| ReadExecuteAccess | 16 | 0 | | | |
| ReadPermissionsAccess | 32 | 0 | | | |
| ChangePermissionsAccess | 64 | 0 | | | |
| FullAccess | 127 | 0 | | | |
| NoPropagation | 128 | 0 | | | |

CfgPersonType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|-------------------|-----------|-------------|--------------------------|-----------------|
| CFGPTNoPersonType | 0 | 0 | 'Unknown Person Type' | Unknown |
| CFGPTPerson | 1 | 0 | 'Person' | Person |
| CFGPTAgent | 2 | 0 | 'Agent' | Agent |

CfgRank

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------------|----|------|-------------------------|----------------------|---|
| CFGNoRank | 0 | 0 | 'Unknown Rank' | Unknown | The value of the corresponding parameter has not changed. |
| CFGUser | 1 | 0 | 'User' | User | User rank. |
| CFGDesigner | 2 | 0 | 'Designer' | Designer | Designer rank. |
| CFGTenantAdministrator | 3 | 0 | 'Administrator' | TenantAdministrator | Tenant Administrator rank. Applies to tenant staff only. |
| CFGServiceAdministrator | 4 | 0 | 'Service Administrator' | ServiceAdministrator | Service Administrator rank. Applies to Service Provider staff only. |
| CFGSuperAdministrator | 5 | 0 | 'Super Administrator' | SuperAdministrator | Super Administrator rank. Applies to Service Provider staff only. |

Comments

Due to the introduction of a flexible access control system in Configuration Server version 5.1.100, the only purpose left for application ranks is to control which functionality of an application is available to the currently logged-on person. The feature requirements for a particular application type govern whether to implement rank-based access to an application's functions, as well as which functions to block or enable for which rank. This information will be made part of the functional specification for that application type. The level of access to Configuration Data objects granted to a particular person does not depend in any way on the set of application ranks for that person.

CfgRecActionCode

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------------|----|------|-------------------------------|------------------|---|
| CFGRACNoRecActionCode | | 0 | 'Unknown Action' | NoRecActionCode | The value of corresponding parameter has not changed |
| CFGRACMarkDB | 1 | 0 | 'No Treatment' | MarkDB | Update call record with call result in database, do not repeat call |
| CFGRACMarkAllChain | | 0 | 'Update all records in chain' | MarkAllChain | Update current record and all record with the same chain ID with call result in database |
| CFGRACCycle | 3 | 0 | 'Redial' | Cycle | In case of specified call result repeat call attempt up to specified number of times. The interval between attempts specified by parameter interval |
| CFGRACRetryIn | 4 | 0 | 'Retry in' | RetryIn | Repeat the call attempt after specified time interval of same day |
| CFGRACRetryAtDate | | 0 | 'Retry at specified date' | RetryAtDate | Repeat the call attempt on specified date |
| CFGRACNextInChain | | 0 | 'Next in chain' | NextInChain | Try next record in a chain |
| CFGRACNextInChainAfter | | 0 | 'Next in chain after' | NextInChainAfter | Try next record in a chain after specified time |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------------|----|------|-----------------------------------|-------------------|--|
| | | | | | interval |
| CFGRACNextInChainAtDate | 0 | 0 | 'Next in chain at specified date' | NextInChainAtDate | Try next record in a chain at specified date/time |
| CFGRACAssignToGroup | 0 | 0 | 'Assign to Group' | AssignToGroup | Update the call record and reschedule the call back to a group of agents |
| CFGRACMarkAsAgentError | 0 | 0 | 'Mark as Agent Error' | MarkAsAgentError | Update the call record with "Agent Error " status |
| CFGRACReschedule | 1 | 1 | 'Reschedule' | Reschedule | |

Comments

The CfgRecActionCode is applicable for Configuration Library/Server release 5.1.5xx only.

RAC in value name means belonging to CfgRecActionCode enumeration.

CfgResourceType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|----------------------------|----|------|--|----------------------------|
| CFGRTNoType | 0 | 0 | 'Unknown' | Unknown |
| CFGRTTimeZone | 1 | 0 | 'Time Zone Resource' | TimeZoneResource |
| CFGRTContract | 2 | 0 | 'Contract Resource' | ContractResource |
| CFGRTSiteAccess | 3 | 0 | 'Site Transfer Resource' | SiteAccessResource |
| CFGRTGVPGroupCustomer | 4 | 0 | 'GVP Customer Group Customer Resource' | GVPGroupCustomerResource |
| CFGRTGVPGroupIVRProfile | 5 | 0 | 'GVP IVRProfile Group IVRProfile Resource' | GVPGroupIVRProfileResource |
| CFGRTGVPServerType6 | 6 | 0 | 'GVP Server Type Resource' | GVPServerType |
| CFGRTGVPServerTypeDisabled | 7 | 0 | 'GVP Server Disabled Type Resource' | GVPServerTypeDisabled |
| CFGRTGVPServer | 8 | 0 | 'GVP Server Resource' | GVPServerResource |
| CFGRTGVPProcessNode | 9 | 0 | 'GVP Process Node Resource' | GVPProcessNodeResource |
| CFGRTGVPDataNode | 10 | 0 | 'GVP Data Node Resource' | GVPDataNodeResource |
| CFGRTGVPTemplate | 11 | 0 | 'GVP Template Resource' | GVPTemplateResource |
| CFGRTGroupHost | 12 | 0 | 'Host Group Host Resource' | GroupHostResource |
| CFGRTTaskGVPIVRProfile | 13 | 0 | 'Task GVP IVRProfile Resource' | TaskGVPIVRProfileResource |
| CFGRTTaskGVPCustomer | 14 | 0 | 'Task GVP Customer Resource' | TaskGVPCustomerResource |
| CFGRTTaskSwitch | 15 | 0 | 'Task Switch Resource' | TaskSwitchResource |
| CFGRTTaskHost | 16 | 0 | 'Task Host Resource' | TaskHostResource |

| Mnemonic | ID | Flag | Short Description | Log Name |
|-----------------|-----------|-------------|------------------------------|-----------------|
| | | | Resource' | |

CfgRouteType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------------------|----|------|-------------------------|--------------------------|---|
| CFGNoRoute | 0 | 0 | 'Unknown Route Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGDefault | 1 | 0 | 'Default' | Default | |
| CFGLabel | 2 | 0 | 'Label' | Label | |
| CFGOverwriteDNIS | 3 | 0 | 'Overwrite DNIS' | OverwriteDNIS | |
| CFGDDD | 4 | 0 | 'DDD' | DDD | |
| CFGIDDD | 5 | 0 | 'IDDD' | IDDD | |
| CFGDirect | 6 | 0 | 'Direct' | Direct | |
| CFGReject | 7 | 0 | 'Reject' | Reject | |
| CFGAnnouncement | 8 | 0 | 'Announcement' | Announcement | |
| CFGPostFeature | 9 | 0 | 'Post Feature' | PostFeature | |
| CFGDirectAgent | 10 | 0 | 'Direct Agent' | DirectAgent | |
| CFGUseExternalProtocol | 11 | 0 | 'Use External Protocol' | UseExternalProtocol | |
| CFGGetFromDN | 12 | 0 | 'Get From DN' | GetFromDN | |
| CFGXRouteTypeDefault | 13 | 0 | 'Default' | XRouteTypeDefault | |
| CFGXRouteTypeRoute | 14 | 0 | 'Route' | XRouteTypeRoute | |
| CFGXRouteTypeDirect | 15 | 0 | 'Direct' | XRouteTypeDirect | |
| CFGXRouteTypeReRoute | 16 | 0 | 'Re-Route' | XRouteTypeReroute | |
| CFGXRouteTypeDirectUII | 17 | 0 | 'Direct UII' | XRouteTypeDirectUII | |
| CFGXRouteTypeDirectANI | 18 | 0 | 'Direct ANI' | XRouteTypeDirectANI | |
| CFGXRouteTypeDirectNoToken | 19 | 0 | 'Direct No Token' | XRouteTypeDirectNoToken | |
| CFGXRouteTypeDNISPooling | 20 | 0 | 'DNIS Pooling' | XRouteTypeDNISPooling | |
| CFGXRouteTypeDirectDNISnANI | 21 | 0 | 'Direct DNIS and ANI' | XRouteTypeDirectDNISnANI | |
| CFGXRouteTypeDirectDigits | 22 | 0 | 'Direct Digits' | XRouteTypeDirectDigits | |
| CFGXRouteTypeForbidden | 23 | 0 | 'Forbidden' | XRouteTypeForbidden | |
| CFGXRouteTypeISCCProtocol | 24 | 0 | 'ISCC defined' | XRouteTypeISCCProtocol | |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------------------|----|------|--------------------------|-------------------------------|------------------|
| | | | protocol' | | |
| CFGXRouteTypePullBack | 13 | 0 | 'PullBack' | XRouteTypePullback | |
| CFGXRouteTypeDirectNetworkCallID | 26 | 0 | 'Direct Network Call ID' | XRouteTypeDirectNetworkCallID | |

Comments

Values from 13 to 25 must be shown on GUI if property Target Type under AccessCodes is set to TargetISCC only, meantime the values from 1 to 12 must be hidden.

For explanations of routing types, refer to the latest version of the *T-Server Developer's Guide*.

CfgScriptType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------------|----|------|-----------------------|----------------------|---|
| CFGNoScript | 0 | 0 | 'Unknown Script Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGDataCollection1 | 1 | 0 | 'Data Collection' | DataCollection | |
| CFGServiceSelection2 | 2 | 0 | 'Service Selection' | ServiceSelection | |
| CFGEnhancedQueueing3 | 3 | 0 | 'Enhanced Queueing' | EnhancedQueueing | |
| CFGScriptConfiguration4 | 4 | 0 | 'Simple Queueing' | SimpleQueueing | |
| CFGSimpleRouting5 | 5 | 0 | 'Simple Routing' | SimpleRouting | |
| CFGEnhancedRouting6 | 6 | 0 | 'Enhanced Routing' | EnhancedRouting | |
| CFGVoiceFile7 | 7 | 0 | 'Voice Data' | VoiceFile | |
| CFGOutboundCampaign8 | 8 | 0 | 'Outbound Campaign' | CFGOutboundCampaign | |
| CFGOutboundFormat9 | 9 | 0 | 'Outbound Format' | CFGOutboundFormat | |
| CFGOutboundList10 | 10 | 0 | 'Outbound List' | CFGOutboundList | |
| CFGOutboundFilter11 | 11 | 0 | 'Outbound Filter' | CFGOutboundFilter | |
| CFGOutboundTreatment12 | 12 | 0 | 'Outbound Treatment' | CFGOutboundTreatment | |
| CFGOutboundAlert13 | 13 | 0 | 'Outbound Alert' | CFGOutboundAlert | |
| CFGSchedule14 | 14 | 0 | 'Schedule' | Schedule | General purpose schedule |
| CFGAlarmDetection15 | 15 | 0 | 'Alarm Detection' | AlarmDetection | A script to provide alarm detection logic |
| CFGAlarmReaction16 | 16 | 0 | 'Alarm Reaction' | AlarmReaction | A script to describe a |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------------|----|------|------------------------------|----------------------|--------------------------------------|
| | | | | | reaction to an alarm |
| CFGVssSystemSchema | 17 | 0 | 'VSS System Schema' | VssSystemSchema | |
| CFGVssSharedSchema | 18 | 0 | 'VSS Shared Schema' | VssSharedSchema | |
| CFGVssServerSchema | 19 | 0 | 'VSS Server Schema' | VssServerSchema | |
| CFGVssObject | 20 | 0 | 'VSS Object' | VssObject | |
| CFGEmailAckReceipt | 21 | 0 | 'E-mail Acknowledge Receipt' | EmailAckReceipt | |
| CFGCapacityRule | 22 | 0 | 'Capacity Rule' | CapacityRule | A script to describe a capacity rule |
| CFGInteractionQueue | 23 | 0 | 'Interaction Queue' | InteractionQueue | |
| CFGInteractionQueueView | 24 | 0 | 'Interaction Queue View' | InteractionQueueView | |
| CFGInteractionWorkBin | 25 | 0 | 'Interaction Work Bin' | InteractionWorkBin | |
| CFGInteractionSubmitter | 26 | 0 | 'Interaction Submitter' | InteractionSubmitter | |
| CFGInteractionSnapshot | 27 | 0 | 'Interaction Snapshot' | InteractionSnapshot | |
| CFGBusinessProcess | 28 | 0 | 'Business Process' | BusinessProcess | |
| CFGSupervisorData | 29 | 0 | 'Supervisor Data' | SupervisorData | |
| CFGGVPDataNode | 30 | 0 | 'GVP Data Node' | GVPDataNode | |

CfgSelectionMode

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------|----|------|------------------------------|-----------------|--|
| CFGSMNoSelectionMode | 0 | 0 | 'Unknown Selection Mode' | Unknown | The value of corresponding parameter has not changed |
| CFGSMByAppDBID1 | 1 | 0 | 'Select By Application' | SelectByAppDBID | An event from application specified by DBID is considered only. See CfgAlarmEvent. |
| CFGSMAppType | 2 | 0 | 'Select By Application Type' | SelectByAppType | |
| CFGSMByAny | 3 | 0 | 'Select By Any' | SelectByAny | Any event. |

Comments

SM in every enumeration value stands to stress the fact it belongs to CfgSelectionMode enumeration.

The CfgSelectionMode specifies the event selection mode that is used for alarm condition analysis.

CfgSolutionType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------------|----|------|-------------------------|-------------------------|--|
| CFGSTNoSolutionType | | 0 | 'Unknown Solution Type' | Unknown | The value of the corresponding parameter has not changed |
| CFGSTDefaultSolutionType | | 0 | 'Default Solution Type' | DefaultSolutionType | The value is used during migration from 5.1.xxx to 5.9 releases. Value is assigned as default for objects of CfgService type that exist in 5.1xx configuration. The value cannot be assigned for new objects of CfgService and CfgSolutionTemplate type. See also description of CfgService and CfgSolutionTemplate. |
| CFGSTIntegratedScreenPop | | 0 | 'Integrated Screen Pop' | IntegratedScreenPop | |
| CFGSTEnterpriseRouting | | 0 | 'Enterprise Routing' | EnterpriseRouting | |
| CFGSTNetworkRouting | | 0 | 'Network Routing' | NetworkRouting | |
| CFGSTOutboundDialing | | 0 | 'Outbound Dialing' | OutboundDialing | |
| CFGSTWorkforceManagement | | 0 | 'Workforce Manager' | WorkforceManagement | |
| CFGSTVoiceSelfService | | 0 | 'Voice Self Service' | VoiceSelfService | |
| CFGSTInternetContactCenter | | 0 | 'Internet Contact' | InternetContactSolution | |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------------------|----|------|--------------------------------|--------------------------|------------------|
| | | | Solution' | | |
| CFGSTWorkflowRouting | | 0 | 'Workflow Routing' | WorkflowRouting | |
| CFGSTInformationAnalysis | | 0 | 'Information Analysis' | InformationAnalysis | |
| CFGSTKnowledgeWorker | | 0 | 'Knowledge Worker' | KnowledgeWorker | |
| CFGSTBranchOffice2 | | 0 | 'Branch Office' | BranchOffice | |
| CFGSTFramework 13 | | 0 | 'Framework' | Framework | |
| CFGSTExpressEdition | | 0 | 'Express Edition' | ExpressEdition | |
| CFGSTMultichannelRouting | | 0 | 'Multimedia' | MultiChannelRouting | |
| CFGSTDesktopNETServerSolution | | 0 | 'Desktop .NET Server Solution' | DesktopNETServerSolution | |

Comments

ST in every enumeration value stands to stress the fact it belongs to CfgSolutionType enumeration.

Please refer to Genesys Enterprise Solutions. Proposed Definitions document for more details regarding each solution type.

CfgStartupType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|---------------------|----|------|------------------------|---------------|---|
| CFGSUTNoStartupType | 0 | 0 | 'Unknown Startup Type' | NoStartupType | The value of corresponding parameter has not changed |
| CFGSUTAutomatic1 | 1 | 0 | 'Automatic' | Automatic | An automatic startup type. The Management Layer controls the application/solution in normal mode i.e., runs and monitors it. |
| CFGSUTManual | 2 | 0 | 'Manual' | Manual | A manual startup type. The solution must be started manually i.e., the Management Layer only can monitor the solution without performing control functions. |
| CFGSUTDisabled | 3 | 0 | 'Disabled' | Disabled | The start of application is disabled i.e., application cannot be started either manually or automatic. |

Comments

SUT in every enumeration value stands to stress the fact it belongs to CfgStartupType enumeration.

The CfgStartupType specifies the mode in which the Application/Solution will be started.

CfgStatDayType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|-------------------|-----------|-------------|--------------------------|-----------------|
| CFGSDTNoType | 0 | 0 | 'Unknown' | Unknown |
| CFGSDTDefault | 1 | 0 | 'Default' | Default |
| CFGSDTDayContract | 2 | 0 | 'Day Contract' | DayContract |

CfgStatTableType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------------------|----|------|--------------------------------|---------------------------|---|
| CFGNoStatTable | 0 | 0 | 'Unknown Stat Table Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGCapacityTable | 1 | 0 | 'Capacity Table' | CapacityTable | |
| CFGQuotaTable | 2 | 0 | 'Quota Table' | QuotaTable | |
| CFGSpecialDayTable | 3 | 0 | 'Special Day Table' | SpecialDayTable | |
| CFGVolumeContractTable | 4 | 0 | 'Volume Contract Table' | VolumeContractTable | |
| CFGVariableRateContractTable | 5 | 0 | 'Variable Rate Contract Table' | VariableRateContractTable | |

CfgSwitchType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------|----|------|--|-------------------------------------|---|
| CFGNoSwitch | 0 | 0 | 'Unknown Switch Type' | Unknown | The value of the corresponding parameter has not changed. |
| CFGNortelMeridian1 | 1 | 0 | 'Nortel Meridian 1' | NortelMeridian | |
| CFGRockwellSpectrum | 2 | 0 | 'Rockwell Spectrum' | RockwellSpectrum | |
| CFGRockwellGalaxy | 3 | 0 | 'Rockwell Galaxy' | RockwellGalaxy | |
| CFGNortelDMS1004 | 4 | 0 | 'Nortel Communication Server 2000/ 2100' | NortelCommunicationServer2000/ 2100 | |
| CFGLucentDefinityECS | 5 | 0 | 'Avaya Communication Manager' | AvayaDefinityECS(MV) | |
| CFGAspectCallCenter | 6 | 0 | 'Aspect CallCenter' | AspectCallCenter | |
| CFGSiemensHicom300E | 7 | 0 | 'Siemens Hicom 300E' | SiemensHicom300E | |
| CFGIntecomIBX80 | 8 | 0 | 'Intecom IBX80' | IntecomIBX80 | |
| CFGEricssonMD110 | 9 | 0 | 'Ericsson MD110' | EricssonMD110 | |
| CFGLucent5ESS | 10 | 0 | 'Lucent 5ESS' | Lucent5ESS | |
| CFGMadge | 11 | 0 | 'Madge' | Madge | |
| CFGNEC | 12 | 0 | 'NEC NEAX' | NEC | |
| CFGFujitsu | 13 | 0 | 'Fujitsu F9600' | Fujitsu | |
| CFGHarrisVoiceFrame | 14 | 0 | 'Teltronics 20-20' | HarrisVoiceFrame | |
| CFGGateway01 | 15 | 0 | 'WorldCom 800 Gateway' | Gateway01 | |
| CFGSiemensHicom150 | 16 | 0 | 'Siemens Hicom 150' | SiemensHicom150 | |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------------------|------|------|--|--------------------------|--|
| CFGSiemensHicom300 | 1700 | 0 | 'Siemens Hicom 300 ACL-H3' | SiemensHicom300 | |
| CFGPhilipsSophoiS3000 | 1800 | 0 | 'Philips Sophoi S3000' | PhilipsSophoiS3000 | |
| CFGMatraMC6500 | 19 | 0 | 'EADS Telecom M6500 Succession' | MatraMC6500 | |
| CFGSiemensHicom150H | 20 | 0 | 'Siemens Hicom 150H' | SiemensHicom150H | |
| CFGEricssonACP1000 | 21 | 0 | 'Ericsson ACP1000' | EricssonACP1000 | |
| CFGSiemensGECiSDX | 22 | 0 | 'Siemens GEC iSDX' | SiemensGECiSDX | |
| CFGAlcatelA4400DHS3 | 23 | 0 | 'Alcatel A4400' | AlcatelA4400DHS3 | |
| CFGGenericSwitch | 24 | 0 | 'Generic Switch' | GenericSwitch | |
| CFGDelcoACD | 25 | 0 | 'Delco ACD' | DelcoACD | |
| CFGHitachiCX8000 | 26 | 0 | 'Hitachi CX8000' | HitachiCX8000 | |
| CFGLGStarex | 27 | 0 | 'LG Starex-ACS' | LGStarex | |
| CFGMitelSX2000 | 28 | 0 | 'Mitel SX-2000' | MitelSX2000 | |
| CFGNortelMeridianCallCenter | 29 | 0 | 'Nortel Communication Server 1000 with SCCS/MLS' | NortelMeridianCallCenter | |
| CFGSiemensHicom150E | 30 | 0 | 'Siemens Hicom 150E' | SiemensHicom150E | |
| CFGSiemensRealitiSDX | 31 | 0 | 'Siemens RealitiSDX iCCL' | SiemensRealitiSDX | |
| CFGTadiranCoral | 32 | 0 | 'Tadiran Coral' | TadiranCoral | |
| CFGVoIPSMCPSwitch | 33 | 0 | 'Voice over IP SMCP Switch' | VoIPSMCPSwitch | |
| CFGVirtualSwitchIIF | 34 | 0 | 'Virtual Switch for IVR In-Front' | VirtualSwitchIIF | The switch for IVR In-Front interface. |
| CFGInternetGateway | 35 | 0 | 'Internet Gateway' | InternetGateway | |
| CFGATT800ICPGateway | 36 | 0 | 'AT&T 800 ICP Gateway' | ATT800ICPGateway | |
| CFGsprintSiteRPGateway | 37 | 0 | 'Sprint SiteRP Gateway' | SprintSiteRPGateway | |
| CFGBellCanadaATFGateway | 38 | 0 | 'Bell Canada ATFGateway' | BellCanadaATFGateway | |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|----------------------------|----|------|--------------------------------------|-------------------------|------------------|
| | | | (Stentor) ATF Gateway' | | |
| CFGAlcatelISCPGateway | 39 | 0 | 'Alcatel SCP Gateway' | AlcatelISCPGateway | |
| CFGBellAtlanticISCPGateway | 40 | 0 | 'Bell Atlantic ISCP Gateway' | BellAtlanticISCPGateway | |
| CFGConcert800Gateway | 41 | 0 | 'Concert 800 Gateway' | Concert800Gateway | |
| CFGAlcatelDTAGSIPGateway | 42 | 0 | 'Alcatel DTAG SCP Gateway' | AlcatelDTAGSIPGateway | |
| CFGKPNNetworkGateway | 43 | 0 | 'KPN Network Gateway' | KPNNetworkGateway | |
| CFGAlcatelTISCPGateway | 44 | 0 | 'Alcatel Telecom Italia SCP Gateway' | AlcatelTISCPGateway | |
| CFGAlcatelBTSCPGateway | 45 | 0 | 'Alcatel BT SCP Gateway' | AlcatelBTSCPGateway | |
| CFG3511ProtocolInterface | 46 | 0 | '3511 Protocol Interface' | 3511ProtocolInterface | |
| CFGDatavoiceDharma | 47 | 0 | 'DataVoice Dharma' | DatavoiceDharma | |
| CFGHuaweiCC08 | 48 | 0 | 'Huawei C_C08' | HuaweiCC08 | |
| CFGLucentIndexSDX | 49 | 0 | 'Avaya INDeX' | LucentIndexSDX | |
| CFGSiemensHicom300H | 50 | 0 | 'Siemens Hicom 300H' | SiemensHicom300H | |
| CFGSiemensHiPath4000 | 51 | 0 | 'Siemens HiPath 4000' | SiemensHiPath4000 | |
| CFGAlcatelA4200 | 52 | 0 | 'Alcatel A4200' | AlcatelA4200 | |
| CFGTenovisIntegral33 | 53 | 0 | 'Tenovis Integral 33' | TenovisIntegral33 | |
| CFGTelera | 54 | 0 | 'Telera' | Telera | |
| CFGNGSN | 55 | 0 | 'NGSN' | NGSN | |
| CFGGenSpec | 56 | 0 | 'GenSpec' | GenSpec | |
| CFGVoicePortal | 57 | 0 | 'Voice Portal' | VoicePortal | |
| CFGKWGateway | 58 | 0 | 'K-Worker Gateway' | K-WorkerGateway | |
| CFGSiemensHicom300Real | 59 | 0 | 'Siemens Hicom 300' | SiemensHicom300Real | |
| CFGGenSpecXML | 60 | 0 | 'GenSpec XML' | GenSpecXML | |
| CFGOPSI | 61 | 0 | 'OPSI' | OPSI | |
| CFGCiscoCM | 62 | 0 | 'Cisco Call Manager' | CiscoCallManager | |

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-----------------------|----|------|------------------------|--------------------|------------------|
| CFGMultimediaSwitch | 68 | 0 | 'Multimedia Switch' | MultimediaSwitch | |
| CFGVerizonISCPGateway | 69 | 0 | 'Verizon ISCP Gateway' | VerizonISCPGateway | |
| CFGAlcatel5020OPSI | 70 | 0 | 'Alcatel 5020 OPSI' | Alcatel5020OPSI | |
| CFGAvayaIPOffice | 66 | 0 | 'Avaya IP Office' | AvayaIPOffice | |
| CFGMitelMN3300 | 67 | 0 | 'Mitel MN-3300' | MitelMN3300 | |
| CFGSamsungIPPCXIAP | 65 | 0 | 'Samsung IP PCX IAP' | SamsungIPPCXIAP | |
| CFGSiemensHiPath3000 | 69 | 0 | 'Siemens HiPath 3000' | SiemensHiPath3000 | |
| CFGEOneQueue | 70 | 0 | 'eOn eQueue' | eOneQueue | |
| CFGTenovisI55 | 71 | 0 | 'Tenovis I55' | TenovisI55 | |
| CFGSIPSwitch | 72 | 0 | 'SIP Switch' | SIPSwitch | |
| CFGDigitroAXS20 | 73 | 0 | 'Digitro AXS/20' | DigitroAXS20 | |
| CFGGVPDIDGroup | 74 | 0 | 'GVP DID Group' | GVPDIDGroup | |
| CFGSIPNetworkSwitch | 75 | 0 | 'SIP Network Switch' | SIPNetworkSwitch | |
| CFGNECSV7000 | 76 | 0 | 'NEC NEAX SV7000' | NEC SV7000 | |
| CFGRadvisionContact | 77 | | | | |
| CFGAvayaTSAPI | 78 | | | | |
| CFGHuaweiNGN | 79 | | | | |
| CFGCiscoUCCE | 80 | | | | |
| CFGMaxSwitchType | 81 | | | | |

CfgTableType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------------|----|------|--------------------------|---------------------|---|
| CFGTTNoTableType@ | | 0 | 'Unknown Table Type' | NoTableType | The value of corresponding parameter has not changed |
| CFGTTCallingList | 1 | 0 | 'Calling List' | CallingList | Calling list |
| CFGTTLogTable | 2 | 0 | 'Log Table' | LogTable | Log Table |
| CFGTTANI | 3 | 0 | 'ANI' | ANI | ANI table |
| CFG TTLATA | 4 | 0 | 'LATA' | LATA | LATA table |
| CFGTTNPA | 5 | 0 | 'NPA' | NPA | NPA table |
| CFGTTNPA_NXX | 6 | 0 | 'NPA-NXX' | NPA-NXX | NPA-NXX table |
| CFGTTStateCode | 7 | 0 | 'State Code' | StateCode | State Code table |
| CFGTTInfoDigits | 8 | 0 | 'Info Digits' | InfoDigits | Info Digits table |
| CFGTTCountryCode@ | | 0 | 'Country Code' | CountryCode | Country Code table |
| CFGTTCustomType10 | | 0 | 'Customer Defined Table' | CustomType | Customer defined table |
| CFGTTDoNotCallList1 | | 0 | 'Do Not Call List' | DoNotCallRegulation | Do not call list table type to support nationwide Do Not Call regulation by OCS |
| CFGTTEmailContactLists | | 0 | 'E-mail Contact List' | EmailContactLists | Email Contact Lists |

Comments

The CfgTableType is applicable for Configuration Library/Server release 5.1.5xx only.

TT in value name means belonging to CfgTableType enumeration.

CfgTargetType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|---------------------------|----|------|---------------------------------|------------------------|--|
| CFGNoTarget | 0 | 0 | 'Unknown Target Type' | NoTarget | The value of the corresponding parameter has not changed. |
| CFGTargetAgent | 1 | 0 | 'Target Agent' | TargetAgent | |
| CFGTargetPlace | 2 | 0 | 'Target Place' | TargetPlace | |
| CFGTargetAgentGroup | 3 | 0 | 'Target Agent Group' | TargetAgentGroup | |
| CFGTargetPlaceGroup | 4 | 0 | 'Target Place Group' | TargetPlaceGroup | |
| CFGTargetRoutingPoint | 5 | 0 | 'Target Routing Point' | TargetRoutingPoint | |
| CFGTargetACDQueue | 6 | 0 | 'Target Queue' | TargetACDQueue | |
| CFGTargetDestinationLabel | 7 | 0 | 'Target Network Destination' | TargetDestinationLabel | |
| CFGTargetACDQueueGroup | 8 | 0 | 'Target Queue Group' | TargetACDQueueGroup | |
| CFGTargetExtRoutingPoint | 9 | 0 | 'Target External Routing Point' | TargetExtRoutingPoint | |
| CFGTargetISCC | 10 | 0 | 'Target ISCC' | TargetISCC | Inter-Server Communication Channel. Must be chosen for External Routing Route types. |

CfgTaskType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|----------------------------|-----------|-------------|---------------------------|-----------------------|
| CFGTTNoType | 0 | 0 | 'Unknown' | Unknown |
| CFGTTUpdateAppIVR | 1 | 0 | 'Update Application IVR' | UpdateApplicationIVR |
| CFGTTRegenerateIVRProfiles | 2 | 0 | 'Regenerate IVR Profiles' | RegenerateIVRProfiles |
| CFGTTRegenerateDIDGroups | 3 | 0 | 'Regenerate DID Groups' | RegenerateDIDGroups |
| CFGTTSwapIVRURLs | 4 | 0 | 'Swap IVR URLs' | SwapIVRURLs |

CfgTraceMode

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|------------------|----|------|------------------------|-------------------|--|
| CFGTMNoTraceMode | | 0 | 'Unknown Trace Mode' | Unknown | The value of corresponding parameter has not changed |
| CFGTMNone | 1 | 0 | 'Trace Is Turned Off' | TraceIsTurnedOff | Trace is turned off |
| CFGTMLocal | 2 | 0 | 'Trace On Client Side' | TraceOnClientSite | A trace on client side only |
| CFGTMRemote | 3 | 0 | 'Trace On Server Side' | TraceOnServerSite | A trace on server side only |
| CFGTMBoth | 4 | 0 | 'Trace On Both Sides' | TraceOnBothSites | A trace on both client and Server sides |

Comments

TM in every enumeration value stands to stress the fact it belongs to CfgTraceMode enumeration.

The CfgTraceMode specifies the connection trace mode that is used between client and server.

CfgTransactionType

Values

| Mnemonic | ID | Flag | Short Description | Log Name | Long Description |
|-------------------------|----|------|----------------------------|-------------------|--|
| CFGTRTNoTransactionType | 0 | 0 | 'Unknown Transaction Type' | Unknown | The value of corresponding parameter has not changed |
| CFGTRTCallData | 1 | 0 | 'Interaction Data' | CallData | A call data |
| CFGTRTBusinessAttribute | 2 | 0 | 'Attribute' | BusinessAttribute | A business attribute |
| CFGTRTBusinessSituation | 3 | 0 | 'Business Rule' | BusinessSituation | A business situation |
| CFGTRTBusinessRule | 4 | 0 | 'Routing Rule' | BusinessRule | A business rule |
| CFGTRTBusinessAction | 5 | 0 | 'Business Action' | BusinessAction | A business Action |
| CFGTRTStatFilter | 16 | 0 | 'Stat Filter' | StatFilter | A statistic filter |
| CFGTRTStatTimeRange | 17 | 0 | 'Stat Time Range' | StatTimeRange | A statistic time range |
| CFGTRTStatTimeProfile | 18 | 0 | 'Stat Time Profile' | StatTimeProfile | A statistic time profile |
| CFGTRTStatType | 19 | 0 | 'Stat Type' | StatType | A statistic type |
| CFGTRTStatMetric | 20 | 0 | 'Statistic' | StatMetric | A set of statistics used by Interaction Router |
| CFGTRTList | 21 | 0 | 'List' | List | A list type |
| CFGTRTMacro | 22 | 0 | 'Macro' | Macro | A macro type |

Comments

TRT in value name means belonging to CfgTransactionType enumeration.

GctiCallState

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|----------------------|----|------|------------------------------|--------------------|
| GctiCStOk | 0 | 0 | 'Ok' | Ok |
| | 1 | 0 | 'Transferred' | |
| | 2 | 0 | 'Conferenced' | |
| GctiCStGeneralError | 3 | 0 | 'General Error' | General Error |
| GctiCStSystemError | 4 | 0 | 'System Error' | System Error |
| | 5 | 0 | 'Remote Release' | |
| GctiCStBusy | 6 | 0 | 'Busy' | Busy |
| GctiCStNoAnswer | 7 | 0 | 'No Answer' | No Answer |
| GctiCStSitDetected | 8 | 0 | 'SIT Detected' | SIT Detected |
| GctiCStAnswMachine | 9 | 0 | 'Answering Machine Detected' | Answering Machine |
| GctiCStAllTrunksBusy | 10 | 0 | 'All Trunks Busy' | All Trunks Busy |
| GctiCStSitInvalidNum | 11 | 0 | 'SIT Invalid Number' | SIT Invalid Num |
| GctiCStSitVacant | 12 | 0 | 'SIT VC (Vacant Code)' | SIT Vacant |
| GctiCStSitIntercept | 13 | 0 | 'SIT IC (Intercept)' | SIT Oper Intercept |
| GctiCStSitUnknown | 14 | 0 | 'SIT Unknown Call State' | SIT Unknown |
| GctiCStSitNocircuit | 15 | 0 | 'SIT NC (No Circuit)' | SIT No Circuit |
| GctiCStSitReorder | 16 | 0 | 'SIT RO (Reorder)' | SIT Reorder |
| GctiCStFaxDetected | 17 | 0 | 'Fax Detected' | Fax Detected |
| | 18 | 0 | 'Queue Full' | |
| | 19 | 0 | 'Cleared' | |
| | 20 | 0 | 'Overflowed' | |
| GctiCStAbandoned | 21 | 0 | 'Abandoned' | Abandoned |
| | 22 | 0 | 'Redirected' | |
| | 23 | 0 | 'Forwarded' | |
| | 24 | 0 | 'Consult' | |
| | 25 | 0 | 'Pickedup' | |
| GctiCStDropped | 26 | 0 | 'Dropped' | Dropped |

| Mnemonic | ID | Flag | Short Description | Log Name |
|-------------------------|----|------|---------------------------|----------------------|
| GctiCStDroppedNoAns | 27 | 0 | 'Dropped on No Answer' | Dropped No Answer |
| GctiCStUnknown | 28 | 0 | 'Unknown Call Result' | Unknown |
| | 29 | 0 | 'Covered' | |
| | 30 | 0 | 'Converse-On' | |
| | 31 | 0 | 'Bridged' | |
| GctiCStSilence | 32 | 0 | 'Silence' | Silence |
| GctiCStAnswer | 33 | 0 | 'Answer' | Answer |
| GctiCStNuTone | 34 | 0 | 'NU Tone' | NuTone |
| GctiCStNoDialTone | 35 | 0 | 'No Dial Tone' | NoDialTone |
| GctiCStNoProgress | 36 | 0 | 'No Progress' | NoProgress |
| GctiCStNoRingBack | 37 | 0 | 'No RingBack Tone' | NoRingBack |
| GctiCStNoEstablished | 38 | 0 | 'No Established Detected' | NoEstablished |
| GctiCStPagerDetected | 39 | 0 | 'Pager Detected' | Pager Detected |
| GctiCStWrongParty | 40 | 0 | 'Wrong Party' | Wrong Party |
| GctiCStDialErr | 41 | 0 | 'Dial Error' | Dial Error |
| GctiCStDropErr | 42 | 0 | 'Call Drop Error' | Call Drop Error |
| GctiCStSwitchErr | 43 | 0 | 'Switch Error' | Switch Error |
| GctiCStNoFreePortErr | 44 | 0 | 'No Port Available' | No Free Port Error |
| GctiCStTransferErr | 45 | 0 | 'Transfer Error' | Transfer Error |
| GctiCStStale | 46 | 0 | 'Stale' | Stale |
| GctiCStAgentCallBackErr | 47 | 0 | 'Agent CallBack Error' | Agent CallBack Error |
| GctiCStGroupCallBackErr | 48 | 0 | 'Group CallBack Error' | Group CallBack Error |
| | 49 | 0 | 'Deafend' | |
| | 50 | 0 | 'Held' | |
| GctiCStDoNotCall | 51 | 0 | 'Do Not Call' | Do Not Call |
| GctiCStCancel | 52 | 0 | 'Cancel Record' | Cancel Record |
| GctiCStWrongNumber | 53 | 0 | 'Wrong Number' | Wrong number |

GctiContactType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|-----------------------------|----|------|---------------------------|---------------------|
| GctiCtTyNoContactType | 0 | 0 | 'No Contact Type' | NoContactType |
| GctiCtTyHomePhone | 1 | 0 | 'Home Phone' | HomePhone |
| GctiCtTyDirectBusinessPhone | 2 | 0 | 'Direct Business Phone' | DirectBusinessPhone |
| GctiCtTyBusinessWithExt | 3 | 0 | 'Business With Extension' | BusinessWithExt |
| GctiCtTyMobile | 4 | 0 | 'Mobile' | Mobile |
| GctiCtTyVacationPhone | 5 | 0 | 'Vacation Phone' | VacationPhone |
| GctiCtTyPager | 6 | 0 | 'Pager' | Pager |
| GctiCtTyModem | 7 | 0 | 'Modem' | Modem |
| GctiCtTyVoiceMail | 8 | 0 | 'Voice Mail' | VoiceMail |
| GctiCtTyPinPager | 9 | 0 | 'Pin Pager' | PinPager |
| GctiCtTyEMail | 10 | 0 | 'E-Mail address' | E-Mail |

GctiRecordStatus

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|-------------------------|-----------|-------------|--------------------------|-----------------|
| GctiRecStNoRecordStatus | 0 | 0 | 'No Record Status' | NoRecordStatus |
| GctiRecStReady | 1 | 0 | 'Ready' | Ready |
| GctiRecStRetrieved | 2 | 0 | 'Retrieved' | Retrieved |
| GctiRecStUpdated | 3 | 0 | 'Updated' | Updated |
| GctiRecStStale | 4 | 0 | 'Stale' | Stale |
| GctiRecStCancelled | 5 | 0 | 'Cancelled' | Cancelled |
| GctiRecStAgentError | 6 | 0 | 'Agent Error' | AgentError |
| GctiRecStChainUpdated | 7 | 0 | 'Chain Updated' | ChainUpdated |
| GctiRecStMissedCallback | 8 | 0 | 'Missed Callback' | MissedCallback |
| GctiRecStChainReady | 9 | 0 | 'Chain Ready' | ChainReady |

GctiRecordType

Values

| Mnemonic | ID | Flag | Short Description | Log Name |
|------------------------------|----|------|------------------------|---------------------|
| GctiRecTyNoRecordType | | 0 | 'No Record Type' | NoRecordType |
| GctiRecTyUnknown | 1 | 0 | 'Unknown Record Type' | Unknown |
| GctiRecTyGeneral | 2 | 0 | 'General' | General |
| GctiRecTyCampaignRescheduled | 3 | 0 | 'Campaign Rescheduled' | CampaignRescheduled |
| GctiRecTyPersonalRescheduled | 4 | 0 | 'Personal Rescheduled' | PersonalRescheduled |
| GctiRecTyPersonalCallBack | 5 | 0 | 'Personal CallBack' | PersonalCallBack |
| GctiRecTyCampaignCallBack | 6 | 0 | 'Campaign CallBack' | CampaignCallBack |
| GctiRecTyNoCall | 7 | 0 | 'No Call' | NoCall |