**UTC – Advanced Condition Services API**

# UtilService – Utility service to check nulls, empty lists, compare lists etc.

* **isNull** – Checks the given value is null and returns true or false as String. Return value is case-sensitive.
  + **Return Value**: String
  + **Parameters**: String or List or Any Value
  + **Example**:
    - call\_function(“UtilService”, “isNull”, “some string”) = “false”
    - call\_function(“UtilService”, “isNull”, call\_function(“UserService”, “getAttribute”, user.id, “FirstName”)) = “false”
* **isEmptySet** – Checks the given list is empty and returns true or false.
  + **Return Value**: String
  + **Parameters**: List
  + **Example**:
    - call\_function(“UtilService”, “isEmptySet”, call\_function(“UserService”, “getAllCountries”, user.id)) = “true”
* **toList** – Splits the given String using a separator and converts to a list of strings.
  + **Return Value**: List of Strings
  + **Parameters**: String (value), String (Separator)
  + **Example**:
    - call\_function(“UtilService”, “toList”, “US,CA,GB”, “,”)
* **getMatchCount** – returns count of common elements in given lists. Lists can be 2 or more.
  + **Return Value**: Number
  + **Parameters**: List 1, List 2, List 3 …
  + **Example**:
    - call\_function(“UtilService”, “getMatchCount”, call\_function(“UserService”, “getLicenses”, user.id), call\_function(“ResourceService”, “getLicenses”, resource.portal.id)) > 0
* **isAnyMatch** – returns true or false if there are any common elements in given lists. Lists can be 2 or more.
  + **Return Value**: String
  + **Parameters**: List 1, List 2, List 3 …
  + **Example**:
    - call\_function(“UtilService”, “isAnyMatch”, call\_function(“UserService”, “getLicenses”, user.id), call\_function(“ResourceService”, “getLicenses”, resource.portal.id))=”true”
* **isAfterCurrentTimeStamp** – Checks the given value is null and returns true or false as String. Return value is case-sensitive.
  + **Return Value**: String
  + **Parameters**: timestamp,timeformat[optional]
  + **Example**:
    - call\_function(“UtilService”, “isAfterCurrentTimeStamp”, “<*timestamp*>”) = “true”

*This is what should be used most of the time – it works on the basis of the default configured datetime format in common\_ac.properties*

* + - call\_function(“UtilService”, “isAfterCurrentTimeStamp”,”<*timestamp*>”,”*<timeformat[dd/MM/yyyyy HH:mm:ss]>*”) = “false”

*This can be used during testing and initial setup in case datetime format being received is different from what is configured*

* **parseMaterial—**parses the SAP material full path to give material name alone
  + **Return Value**: String
  + **Parameters**: SAPMaterialFullName
  + **Example**:
    - call\_function(“UtilService”, “**parseMaterial**”, resource.object.id)
    - call\_function(“AuthorityService”, “getCount”, “aim.ItemNo =?”, call\_function(“UtilService”, “**parseMaterial**”, resource.object.id))>1

# UserService – to query user’s data

* **getAttribute** – returns value of one given attribute
  + **Return Value**: String
  + **Parameters**: User ID (String), Attribute Name (String)
  + **Example**:
    - call\_function(“UserService”, “getAttribute”, user.id, “FirstName”) = “Sam”
  + Allowed attribute names are; FirstName, LastName, USPerson, CompanyId, CGDESS, CGDEExpiration, UserType, UserID, WindowsSID, CountryOfBirth, CountryOfCompany, CountryOfCitizenship, CountryOfSecondaryCitizenship
  + Note: All attribute names are case-sensitive.
* **getAllCountries** – returns unique list of all country attributes values
  + **Return Value**: List of Strings
  + **Parameters**: User ID (String), Type of Country List (such as ITAR, EAR, ECL), User Physical Location
  + **Example**:
    - call\_function(“UserService”, “getAllCountries”, user.id, “ITAR”, user.physicalLocation)
  + If the user has US, CA, US, GB as countries of birth, company, citizenship and secondary citizenships respectively, then this function returns only US, CA, GB.
  + Since it is a list, we cannot compare the result directly in the advanced condition text area of Policy Studio. So, we can use UtilService if needed. i.e. call\_function(“UserService”, “getAllCountries”, user.id) = call\_function(“UtilService”, “toList”, “US,GB,CA”, “,”)
  + If the user.physicalLocation is non-empty, then it will be added to the result of call\_function.
  + List of Countries to include for ITAR, EAR or ECL can be configured in Common\_AC.properties – towards end of the file.
* **getLicenses** – returns unique list of all Authority IDs (internal primary key) of this user. Licenses are combined from both LDAP licenses attribute (pipe | separated list) and BizAuthDB’s AuthorityUser table
  + **Return Value**: List of Strings
  + **Parameters**: User ID (String)
  + **Example**:
    - call\_function(“UserService”, “getLicenses”, user.id)
* **getInternalLicenseNos** – returns unique list of all Internal License Nos (as specified in BizAuthDB) of this user. Licenses are combined from both LDAP licenses attribute (pipe | separated list) and BizAuthDB’s AuthorityUser table
  + **Return Value**: List of Strings
  + **Parameters**: User ID (String)
  + **Example**:
    - call\_function(“UserService”, “getInternalLicenseNos”, user.id)

# ResourceService – to query resource’s data, for now licenses only.

* **getLicenseNos** – returns unique list of all Authority IDs (internal primary key) of this resource. Licenses are from BizAuthDB’s AuthorityResource table
  + **Return Value**: List of Strings
  + **Parameters**: Resource ID (String)
  + **Example**:
    - call\_function(“ResourceService”, “getLicenseNos”, resource.portal.id)
* **getInternalLicenseNos** – returns unique list of all Internal License Nos (as specified in BizAuthDB) of this resource Licenses are from BizAuthDB’s AuthorityResource table
  + **Return Value**: List of Strings
  + **Parameters**: Resource ID (String)
  + **Example**:
    - call\_function(“ResourceService”, “getInternalLicenseNos”, resource.portal.id)
* **getValidLicenseNos** – returns unique list of all Authority IDs (internal primary key) of this resource. Licenses are from BizAuthDB’s AuthorityResource table, **that are valid and active.**
  + **Return Value**: List of Strings
  + **Parameters**: Resource ID (String)
  + **Example**:
    - call\_function(“ResourceService”, “getValidLicenseNos”, resource.portal.id)
* **getValidInternalLicenseNos** – returns unique list of all Internal License Nos (as specified in BizAuthDB) of this resource Licenses are from BizAuthDB’s AuthorityResource table, **that are valid and active.**
  + **Return Value**: List of Strings
  + **Parameters**: Resource ID (String)
  + **Example**:
    - call\_function(“ResourceService”, “getValidInternalLicenseNos”, resource.portal.id)

# AuthorityService – to query licenses based on a given criteria

* **getLicenseNos** – returns unique list of all Authority IDs (internal primary key) for the given criteria.
  + **Return Value**: List of Strings
  + **Parameters**: Selection criteria (String), Param 1, Param 2, Param 3 … Number of parameters depend on the place-holders in criteria.
  + **Example**:
    - call\_function(“AuthorityService”, “getLicenseNos”, “jcm.Jurisdiction = ? and afsm.ForeignConsignee = ? and ? in actm.CountryCode”, resource.portal.jurisdiction, call\_function(“UserService”, “getAttribute”, user.id, “CompanyId”), call\_function(“UserService”, “getAttribute”, user.id, “CountryOfBirth”) )
  + In the criteria, the column names should follow a convention using alias names. Below are list of valid column names. Note: All alias/column names are case-sensitive.
    - au.Scope
    - au.Sme
    - au.Platform
    - au.Model
    - au.InternalLicenseNo
    - aim.ItemNo
    - acm.Characteristic
    - jcm.Jurisdiction
    - jcm.Classification
    - actm.CountryCode
    - afsm.ForeignConsignee
    - aaum.WindowsSID
    - aaum.UserId
    - arm.Resource
* **getInternalLicenseNos** – returns unique list of all Internal License Nos (as specified in BizAuthDB) for the given criteria.
  + **Return Value**: List of Strings
  + **Parameters**: Selection criteria (String), Param 1, Param 2, Param 3 …
  + **Example**: Refer above.
* **getValidLicenseNos** – returns unique list of all Authority IDs (internal primary key) for the given criteria, **that are valid and active.**
  + **Return Value**: List of Strings
  + **Parameters**: Selection criteria (String), Param 1, Param 2, Param 3 …
  + **Example**: Refer above.
* **getValidInternalLicenseNos** – returns unique list of all Internal License Nos (as specified in BizAuthDB) for the given criteria, **that are valid and active.**
  + **Return Value**: List of Strings
  + **Parameters**: Selection criteria (String), Param 1, Param 2, Param 3 …
  + **Example**: Refer above.
* **getCount** – returns count of all licenses for the given criteria.
  + **Return Value**: Number
  + **Parameters**: Selection criteria (String), Param 1, Param 2, Param 3 …
  + **Example**: Refer above.
* **getValidCount** – returns count of all licenses for the given criteria, **that are valid and active.**
  + **Return Value**: Number
  + **Parameters**: Selection criteria (String), Param 1, Param 2, Param 3 …
  + **Example**: Refer above.
* **getCCLCountries** – returns CCL Country Codes list for the given parameters.
  + **Return Value**: List of Strings
  + **Parameters**: Jurisdiction (String), Classification (String), Reason for control (String). Reason for control is an optional parameter and if given, it should be a proper reason for control code.
  + **Example**:
    - call\_function(“AuthorityService”, “getCCLCountries”, resource.portal.jurisdiction, resource.portal.classification, “AT”)