

# connector-ldap userParameters extension

## Introduction

This page is the documentation of an extension implemented in the [LDAP-Connector developed by evolveum](#). The extension enables the connector to read and write the userParameters Active Directory attribute.

This abstract will not go in depth into the purpose of every single attribute of userParameters. Please consider the [Microsoft documentation](#) for that.

## UserParameters structure

The userParameters AD attribute is a container attribute for a number of different information. It contains attributes of different types, stored as a binary string. For more information about the structure of userParameters visit the [Microsoft documentation](#).

## Attribute types

The attributes stored in userParameters have different types:

- **String**: Variable-length ASCII character array
- **Integer**: either 32-bit unsigned int or 8 bit signed int

## CtxCfgFlags1

This attribute of userParameters is again a container containing numerous boolean flags. Each flag is represented by a single bit in the container which is represented as unsigned integer in the end. The int value changes each time a flag is (un-)set. For more information about the purpose of each flag please consider the [Microsoft documentation](#).

## Implementation

### Configuration

To enable this feature the configuration parameter **rawUserParametersAttribute** has to be set to **false**. If set to **true** the userParameters attribute will not be parsed into its sub-attributes and will not be editable.

If activated the connector schema will be extended by the sub-attributes of userParameters including the flags of CtxCfgFlags1.

### Attribute values

The following will list the attribute values that are supported for each attribute:

#### Boolean values:

Represented as Booleans.

#### Integer Values

Since not all ICF-API-Implementations support Integer values, all integer attributes of userParameters have to be represented as string when given to the Connector. When returned from the connector they will also be represented as Strings.

#### String Values

String values are represented by any UTF-16 encoded String of variable length.

#### Wide Strings

For Each string value there is a wide string representation stored in the userParameters. This means if a string value is edited the wide string value is edited as well and vice versa.

## CTX\_SHADOW

This attribute can only have the following (case sensitive!) String values:

- **Disable**

- **EnableInputNotify**
- **EnableInputNoNotify**
- **EnableNoInputNotify**
- **EnableNoInputNoNotify**

Any other value is inserted here will cause an error.

## Time Values

Time values are represented as unsigned integer. The number means a duration in **Minutes**. Affected attributes are *CtxMaxConnectionTime*, *CtxMaxDisconnectionTime* and *CtxMaxIdleTime*.

## Testcases

To test the Connector especially regarding the CtxFlags1 attribute:

### TC1

1. Create user in midPoint without any userParameter attribute
2. Add AD projection
3. Modify user in midPoint and set a userParameter attribute
4. check in AD if attribute is present

### TC2

1. Create user in midPoint with userParameter attribute
2. Add AD projection
3. check in AD if attribute is present

### TC3

1. Create user in AD which is not present in midPoint
2. search account through midPoint Resource
3. Set userParameter attribute through shadow
4. Check in AD if attribute is present

### TC4

1. Create user in AD which is not present in midPoint
2. Set userParameter attribute in AD
3. search account through midPoint Resource
4. Set userParameter attribute through shadow
5. Check in AD if attribute is present