

# Besturingssystemen III – LDAP

Bert De Saffel

Master in de Industriële Wetenschappen: Informatica Academiejaar 2018–2019

Gecompileerd op 30 december 2018

△ Importeren type library:

```
use Win32::OLE::Const 'Active DS Type Library';
```

△ Binden van een object:

```
sub bind_object {
    my $parameter = shift;
    my $moniker, $object;

    # ||| INLOGPROCEDURE
    # thuis   : 193.190.126.71
    # school  : 192.168.16.16
    my $ip = "193.190.126.71";
    $moniker = "LDAP://$ip/" . $parameter;
    my $login = "Bert De Saffel";
    my $pass = "Bert De Saffel";
    my $dso = Win32::OLE->GetObject("LDAP:");

    $object = $dso->openDSObject(
        $moniker, $login, $pass, ADS_SECURE_AUTHENTICATION);

    print Win32::OLE->LastError() if Win32::OLE->LastError();
    return $object;
}
```

△ Printen attribuut:

```
my $object = bind_object(...);
print $object->{attribuut};
```

△ Op een klasse filteren in een container:

```
my $object = bind_object(...);
$object->{Filter} = [ 'user' ];
for (in $object){ # zal enkel AD-objecten bevatten van type 'user'
    ...
}
```

△ Property Cache initialiseren (belangrijk bij binding RootDSE)

```
my $root = bind_object('RootDSE');
$root->GetInfo();

#Je kan het ook combineren op 1 lijn

(my $root = bind_object('RootDSE'))->GetInfo();
```

△ Verschillende contexts:

- △ DefaultNamingContext
- △ ConfigurationNamingContext
- △ SchemaNamingContext

△ Belangrijke functies:

- △ \$Object->Get("attribuut") # equivalent met \$Object->{attribuut}

- △ `$Object→GetEx()` # bedoeld voor multivalued attributen te verwerken, geeft altijd een referentie naar een array terug, ook al is er maar één waarde.
- △ `$Object→GetInfo()` # Voor de meeste attributen te initialiseren.
- △ `$Object→GetInfoEx([referentie naar een array van attributen], 0)` # expliciet zeggen welke properties moeten geïnitieerd worden
- △ `$Object→SetInfo()`

△ Gebruik maken van **IADsPropertyList**:

```
my $object = bind_object (...);
$object->GetInfo();
while(my $attribuut = $object->Next()){
    # toegang tot Name, ADsType en Values attributen van $attribuut
    # ,dat eigenlijk een instantie is van IADsPropertyEntry.
    for my $waarde (@{$attribuut->{Values}}){
        # Heel rare manier om de waarde van $waarde op te halen
        # verwacht het ADsType dat $waarde zou moeten bevatten
        # Elke waarde van $attribuut->{Values} IS van hetzelfde type
        # Meestal is er ook slechts maar 1 waarde
        print $waarde->GetObjectProperty($attribuut->{ADsType});
    }
}
```

△ In schemacontext kan het handig zijn om filteren op attributeSchema of classSchema:

```
my $root = bind_object('RootDSE');
my $schema = bind_object($root->Get("schemanamingcontext"));
$schema->{Filter} = ["attributeSchema"];
# OF
# $schema->{Filter} = ["classSchema"];
```

△ Enumereren met *ADODBConnection* en *ADODBCommand*:

```
my $connection = Win32::OLE->new("ADODB.Connection");
$connection->{Provider} = "ADsDSOObject";
$connection->{Properties}->{"User ID"} = "Bert De Saffel";
$connection->{Properties}->{"Password"} = "Bert De Saffel";
$connection->{Properties}->{"Encrypt Password"} = 1;
$connection->Open();
my $command = Win32::OLE->new("ADODB.Command");
$command->{ActiveConnection} = $connection;
$command->{Properties}->{"Page Size"} = 20;
$command->{CommandText} = "... " # LDAP query
my $resultset = $command->Execute();
until($resultset->{EOF}){
    print $resultset->Fields("... ")->{Value};
    $resultset->MoveNext();
}
$resultset->Close();
```

△ Objectcategory waarden:

- 0 abstracte klasse
- 1 structurele klasse
- 2 abstracte klasse
- 3 hulpklassen