**Get function**

The GET function in Dynamics NAV (now Dynamics 365 Business Central) is used to retrieve a single record from a table based on the values of the primary key fields. This function is particularly useful when you need to fetch a specific record and you know the primary key values.

Syntax:

|  |
| --- |
| Record.GET([Value],...) |

**Parameters**

**Record**: The record variable representing the table from which you want to retrieve the record.

**Value**: The value(s) of the primary key fields that you want to use to find the specific record.

Let’s create a table called ‘Students’.

|  |
| --- |
|  |

Then we enter the following data.

|  |
| --- |
|  |

Open codeunit from object designer then create a global variable there.

|  |
| --- |
|  |

I name the variable ‘studentRec’ with data type ‘Record’ and in subtype I linked the table we created i.e. ‘Students’.

Then in codeunit, I wrote the following code then saved it and ran it.

|  |
| --- |
|  |

|  |
| --- |
|  |

**Output**

**Usage Notes**

* **Primary Key Fields:** The GET function requires the values of all primary key fields to locate the record. If the table has multiple primary key fields, you need to pass all those values in the correct order.
* **Performance:** The GET function is efficient because it uses the primary key index to locate the record directly.
* Error Handling: It's good practice to handle the scenario where the record is not found by using an IF statement, as shown in the examples.

**SetRange Function**

The SETRANGE function in Dynamics NAV is used to set a filter on a field in a record variable to a specific range of values. This is particularly useful when you want to work with a subset of records that meet certain criteria.

|  |
| --- |
| Record.SETRANGE(Field, [FromValue], [ToValue]) |

**Parameters**

* **Record:** The record variable representing the table on which you want to set the filter.
* **Field:** The field on which you want to set the filter.
* **FromValue (optional):** The starting value of the range.
* **ToValue (optional):** The ending value of the range.

Again, we will create a new codeunit and just like before we create a local variable with name ‘studentRec’ then we put the following codes.

|  |
| --- |
|  |

Finally run it after saving it.

|  |
| --- |
|  |

**Explanation**

**RESET:** The RESET function clears any existing filters on the record variable.

**SETRANGE:** The SETRANGE function sets a filter on the Fee field for values between 1000 and 3000.

**FINDFIRST:** The FINDFIRST function tries to find the first record that matches the filter.

**REPEAT-UNTIL Loop:** If a matching record is found, the loop displays the details of each matching student. If no matching record is found, a message is displayed indicating that no students were found within the specified fee range.

**Usage Notes**

**Single Value Filter:** If you want to filter by a single value, you can omit the ToValue parameter. For example, to find students with a fee of exactly 1500:

studentRec.SETRANGE(Fee, 1500);

**Clearing Filters:** If you need to clear a filter that was previously set using SETRANGE, you can call RESET again on the record variable.

**SetFilter**

The SETFILTER function in Dynamics NAV is used to apply a filter to a field in a record variable using an expression. This function provides more flexibility compared to SETRANGE, allowing you to use complex expressions and wildcards.

|  |
| --- |
| Record.SETFILTER(Field, String [, Value1, ...]) |

**Parameters**

**Record:** The record variable representing the table on which you want to set the filter.

**Field:** The field on which you want to set the filter.

**String:** The filter expression, which can include placeholders (%1, %2, etc.) for values.

**Value1, ...:** The values to be inserted into the placeholders in the filter expression.