|  |  |
| --- | --- |
| **PRODUCT:** | Plant SCADA |
| **CATEGORIES:** | Usability |
| **MARKET:** | All |
| **PROBLEM OR NEED TO SOLVE:** | Static maps in Plant SCADA Studio for instance used for enumerations |
| **EXPECTED FUNCTIONALITY:** | See next pages |
| **RATING** | 8 |
| **BUSINESS CASE** |  |
| **REQUESTOR(S)** | Site ID:85563  Site Name : Rijkswaterstaat Heinenoordtunnel |
|  | If known, mentioned related existing enhancement requests.  (**To be entered by Tech Support Engineer**) |

# Static Maps in Plant SCADA Studio

Projects often have the need for enumerations. Variable Tag values represent for instance a state. Especially in tunnel projects there are a lot of different enumerations. Currently you who have to script all these enumerations in Cicode and create maps on startup in runtime. It would make things easier if you could define these maps in the Plant SCADA Studio instead of scripting this. Here’s an example of how this could work:

## Define a map

A screenshot of a computer

Description automatically generated

## Define the keys and values of each map

A screenshot of a computer

Description automatically generated

## For a variable tag add another field in which you can define the map:

A screenshot of a computer

Description automatically generated

## Compiler:

* Check if mapname is unique
* Check if mapnames and all keys don’t contain spaces
* Check if map defined at Variable Tag exists
* Check if all keys of one map are unique
* Write all maps, keys and values into the \_maps.rdb

## At startup runtime:

* Open the \_maps.rdb
* Create all defined maps
* Add all keys and values into the maps

In runtime it can look like (in this case wrapped in a combobox):

A close up of a sign

Description automatically generated

## Cicode:

Modify:

* TagGetProperty() and AssGetProperty() functions, add a Property with name Map which returns the name of the defined map.
* TagInfo (), TagInfoEx(), AssInfo (), AssInfoEx() functions, add a Type which returns the name of the defined map.

Add:

* SubscriptionGetMap(iHandle, sOffset ), which returns the value from the map

Create a new function which returns the value from the map based on the value of the tag, something like:

STRING

FUNCTION

GetMapValue(STRING sTag, INT iKeyName)

STRING sMap;

STRING sReturn;

STRING sKeyName;

sMap = TagGetProperty(sTag, ”Map”);

IF sMap <> “” THEN

IF MapExists(sMap) THEN

sKeyName = IntToStr(iKeyName);

IF MapKeyExists(sMap, sKeyName) THEN

sReturn = MapValueGet(sMap, sKeyName);

ELSE

sReturn = “ERROR : “ + sKeyName + “ doesn’t exist in map” + sMap;

END

ELSE

sReturn = “ERROR : “ + sMap + “ doesn’t exist”;

END

ELSE

sReturn = “ERROR : No Map defined for tag : “ + sTag;

END

RETURN sReturn;

END

PUBLIC

INT

FUNCTION

AddMapsOnStartup()

STRING sMap, sKey, sValue;

INT hRDB;

INT iReturn;

hRDB = RdbOpen("\_Maps");

IF (hRDB <> -1) THEN

miError = RdbFirstRec(hRDB);

WHILE (miError <> -1) DO

sMap = RdbGet(hRDB, "Map");

sKey = RdbGet(hRDB, "Key");

sValue = RdbGet(hRDB, "Value");

iReturn = iReturn + \_AddMapKeyValue(sMap, sKey, sValue);

miError = RdbNextRec(hRDB);

SleepMS(0);

END

RdbClose(hRDB);

ELSE

iReturn = -1;

END

RETURN iReturn;

END

PRIVATE

FUNCTION

\_AddMapKeyValue(STRING sMap, STRING sKey, STRING sValue)

INT iReturn = -1;

IF sMap <> “” AND sKey <> “” AND sValue <> “” THEN

sMap = MapOpen(sMap, 2);

iReturn = MapValueSet(sMap, sKey, sValue, 2)

END

RETURN iReturn

END

To be able to show a dropdrown menu like this:

A screenshot of a computer

Description automatically generated

The following Cicode can be used

PUBLIC

INT

FUNCTION

ShowMapMenu(STRING sTag, STRING sFont = "")

INT nX;

INT nY;

INT iSelection;

INT nAN = KeyGetCursor();

INT nValue;

INT nMenuID = 1;

INT iReturn = -2;

STRING sMap;

STRING sKey;

STRING sMenuValueLookupMap;

ErrSet(1);

IF StrToInt(UserInfo(0)) = 1 THEN

IF sTag <> "" THEN

sMap = TagGetProperty(sTag, ”Map”);

sMenuValueLookupMap = MapOpen("", 0);

sKey = MapKeyFirst(sMap);

WHILE(sKey <> "") DO

DspPopupMenu(0, MapValueGet(sMap, sKey));

MapValueSet(sMenuValueLookupMap, nMenuID, sKey);

sKey = MapKeyNext(sMap, sKey);

nMenuID = nMenuID + 1;

END

nX = DspGetAnLeft(nAN);

nY = DspGetAnBottom(nAN);

iSelection = \_ShowMapMenu("", nX, NY, TRUE, TRUE, sFont);

IF iSelection > 0 THEN

nValue = MapValueGet(sMenuValueLookupMap, iSelection);

iReturn = TagWrite(sTag, nValue, 0, 1);

END

MapClose(sMenuValueLookupMap);

END

ELSE

iReturn = -1;

END

ErrSet(0);

RETURN iReturn;

END

PRIVATE

INT

FUNCTION

\_ShowMapMenu(STRING sData = "", INT iXPos = -1, INT iYPos = -1, INT bScaleX = TRUE, INT bScaleY = TRUE, STRING sFont = "")

INT nTime;

INT nWindow;

INT nSelection;

INT iReturn;

nTime = TimeCurrent();

nWindow = WinNumber();

iReturn = PageSetInt(mcsPopupMenuTimestamp, nTime, nWindow);

IF iReturn = 0 THEN

iReturn = \_MapSetFont(sFont);

nSelection = DspPopupMenu(-1, sData, iXPos, iYPos, bScaleX, bScaleY);

END

IF (WinNumber() <> nWindow) THEN

nSelection = -1;

ELSE

IF (PageGetInt(mcsPopupMenuTimestamp, nWindow) <> nTime) THEN

nSelection = -1;

END

END

RETURN nSelection;

END

PRIVATE

INT

FUNCTION

\_MapSetFont(STRING sFont)

INT hFont;

INT iReturn = -1;

IF sFont = "" THEN

hFont = DspFontHnd(GetFontMultiRes(WndMonitorIsUHD4K(), "SA\_Menu"));

ELSE

hFont = DspFontHnd(sFont);

END

IF (hFont <> -1) THEN

DspSetPopupMenuFont(hFont);

iReturn = 0;

END

RETURN iReturn;

END

# AVEVA Plant SCADA Historian connector

Add the map defined at the Variable Tag into the Historian configuration. Import all the maps into the Historian configuration. The value of a tag in the Historian configuration can be translated into the value defined in the map.