

Managing plug-in resources

RestoreLastState

restores the last state of the plug-in (for example, last loaded map).

```
void RestoreLastState()
```

FreeState

Deallocates plug-in resources when it is not used by the program (for example, fully unloads a map from the memory).

```
void FreeState()
```

Взаимодействие с меню SmartNavigator - a

MenuGetItemsCount

Returns a number of necessary menu items

```
int MenuGetItemsCount();
```

Return value

Returns a number of necessary menu items to be used by the plug-in

MenuGetItemName

Sets menu item name

```
int MenuGetItemName(  
    int index,  
    wchar_t * name,  
    int length,  
    bool * checked );
```

Параметры

- **index** – number of the item in the menu
- **name** – item name
- **length** – number of askable characters in the field **name**
- **checked** – indicates that a menu item can be checked

Return value

Returns a number of characters from the **name**.

MenuDoAction

Handles corresponding menu items.

```
bool MenuDoAction(  
    int index,  
    void* hwnd )
```

Parameters

- index – number of the selected menu item
- hwnd – parent window's HANDLE (in case you need to create a dialog box)

Return value

Returns **true** if plug-in captures the canvas for map drawing (i.e. when a plug-in becomes active for map drawing) and returns **false** in other cases.

Transformation

TransformationSet

Sets current transformation settings.

```
bool Transformation (  
    double longitude,  
    double latitude,  
    double scale,  
    double direction,  
    double screen_width,  
    double screen_height );
```

Parameters

- longitude – longitude (measure off degrees) от -180.0 до 180.0
- latitude – latitude (measure off degrees) от -90.0 до 90.0
- scale – scale (measure off meters in a pixels)
- direction – bending of the direction to the north (measure off radians, clockwise)
- screen_width – width of a map window (measure off pixels)
- screen_height – hight of a map window (measure off pixels)

Return value

returns **true** if the transformation has been successfully performed and returns **false** if the transformation with these parameters is impossible or has not been performed because of any other reasons

Remark

Set longitude and latitude are placed in the center of the screen by default.

TransformationGet

Returns parameters of the current transformation.

```
void TransformationGet(  
    double* longitude,  
    double* latitude,  
    double* scale,  
    double* direction );
```

Parameters

- longitude – longitude (measure off degrees) от -180.0 до 180.0
- latitude – latitude (measure off degrees) от -90.0 до 90.0
- scale – scale (measure off meters in a pixels)
- direction – bending of the direction to the north (measure off radians, clockwise)

TransformationConvertToScreen

Transforms geographical coordinates into screen coordinates.

```
bool TransformationConvertToScreen(  
    double longitude,  
    double latitude,  
    double* screen_x,  
    double* screen_y );
```

Параметры

- longitude – longitude (measure off degrees)
- latitude – latitude (measure off degrees)
- screen_x – position on the screen (measure off pixels)
- screen_y – position on the screen (measure off pixels)

Return value

Returns **true** if transformation has been completed successfully.

Remark

screen_x = 0; screen_y = 0 – left upper corner.

TransformationConvertFromScreen

Transforms screen coordinates into geographical coordinates.

```
bool TransformationConvertFromScreen(  
    double screen_x,  
    double screen_y,  
    double* longitude,  
    double* latitude );
```

Parameters

- screen_x – position on the screen (measure off pixels)
- screen_y - position on the screen (measure off pixels)
- longitude – longitude (measure off degrees)
- latitude – latitude (measure off degrees)

Return value

Returns **true** if transformation has been completed successfully.

Remark

screen_x = 0; screen_y = 0 – left upper corner.

TransformationGetScale

Aligning upper bound and lower bound.

```
double TransformationGetScale(  
    double scale,  
    int direction );
```

Parameters

- scale – askable scale
- direction – method of scale aligning
 - 1 – the largest scale that is smaller than the askable scale is chosen
 - 0 – the scale that is most close to the askable scale is chosen
 - 1 – the smallest scale that is bigger than the askable scale is chosen

Return value

Returns aligned scale

Remark

If there is no scale to solve a query found, then the largest or smallest is returned depending on the query.

Map

GetBoundingBox

Returns coordinates of the bounding box of the displayed map

```
void GetBoundingBox(  
    double* longitude_min,  
    double* latitude_min,  
    double* longitude_max,  
    double* latitude_max );
```

Parameters

- longitude_min – min longitude of the bounding box (measure off degrees)
- latitude_min - min longitude of the bounding box (measure off degrees)
- longitude_max - max longitude of the bounding box (measure off degrees)
- latitude_max – max longitude of the bounding box (measure off degrees)

Remark

Return value is used for scaling as a center of the screen when the map is fully visible on the screen.

If a map doesn't have distinguished boundaries (e.g. the globe) or when you don't want to use auto scaling, it is necessary to set all values to zero.

MapDraw

Draws the map

```
void MapDraw(  
    void* hdc );
```

Parameters

- hdc – HDC for drawing the map, the size of the screen is contained in the function TransformationSet