

SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

Applies To:

SAP 4.6c and above.

Summary

This is tiny yet useful ABAP class, implementing progress indicator functionality by utilizing SAPGUI_PROGRESS_INDICATOR function module.

Sometimes you need to show a progress of some process. For example, you can process quite large internal table in a loop. Also it should be better to show progress only when it really changes and not very often or else a user can see blinking watch-icon in the GUI status line. Also it would be nice not to bother with percent calculation. All this can be done with ZCL_PROGRESS_INDICATOR class.

First, call its START method to define total amount of process steps (say internal table line count) and status line text. Also you can set percent threshold which defines frequency of progress indicator updates. If you set threshold to 5 then the progress indicator will be updated less often than 5% of step count.

Next, call ADVANCE method on every countable process step (say on every loop step). It will show progress when needed and increment internal progress counts. Also you can call SET method to set current progress absolutely.

Finally, call FINISH method to clear GUI status line.

By: Sergei Korolev

Company: IBS company

Date: 24 June 2005

Class declaration

```
CLASS zcl_progress_indicator DEFINITION
  PUBLIC
  CREATE PUBLIC .

  *** public components of class ZCL_PROGRESS_INDICATOR
  *** do not include other source files here!!!
  PUBLIC SECTION.

  DATA progress_text TYPE string .
  DATA high_bound TYPE i .

  METHODS start
    IMPORTING
      value(i_status_text) TYPE any OPTIONAL
      value(i_high_bound) TYPE i DEFAULT 100
      value(i_percent_treshold) TYPE i DEFAULT 5 .
  METHODS finish .
  METHODS advance
    IMPORTING
      value(i_status_text) TYPE any OPTIONAL
      value(i_step) TYPE i DEFAULT 1 .
  METHODS set
    IMPORTING
      value(i_status_text) TYPE any OPTIONAL
```

```
        value(i_progress) TYPE i .

*** private components of class ZCL_PROGRESS_INDICATOR
*** do not include other source files here!!!
PRIVATE SECTION.

    DATA current_progress TYPE i .
    DATA last_showed_progress TYPE i .
    DATA treshold TYPE i .
    data TRESHOLD type I .

ENDCLASS.
```

Class implementation

```
CLASS class IMPLEMENTATION.

METHOD start .
    CLEAR:
        current_progress,
        last_showed_progress.

    treshold = i_percent_treshold.
    progress_text = i_status_text.
    high_bound = i_high_bound.
    IF high_bound > 0.
        last_showed_progress = 1.
    ENDIF.

    CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR'
        EXPORTING
            percentage = last_showed_progress
            text       = i_status_text.

ENDMETHOD.

METHOD FINISH .
    CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR'.
ENDMETHOD.

METHOD advance .
    DATA:
        l_progress TYPE i.

    l_progress = current_progress + i_step.
    CALL METHOD set( i_status_text = i_status_text
        i_progress = l_progress ).
ENDMETHOD.

METHOD set .
    DATA:
        text TYPE string,
        delta TYPE i,
        l_new_progress TYPE i.

    current_progress = i_progress.
    IF current_progress > high_bound.
        current_progress = high_bound.
    ENDIF.
```

```
l_new_progress = 100 * current_progress / high_bound.  
delta = ABS( l_new_progress - last_showed_progress ).  
IF delta >= treshold.  
    last_showed_progress = l_new_progress.  
    IF i_status_text IS INITIAL.  
        text = progress_text.  
    ELSE.  
        text = i_status_text.  
    ENDIF.  
    CALL FUNCTION 'SAPGUI_PROGRESS_INDICATOR'  
        EXPORTING  
            percentage = last_showed_progress  
            text       = text.  
ENDIF.  
ENDMETHOD.  
  
ENDCLASS.
```

Usage demo

This demo shows most simple way of the class usage. It is assumed that the class created as a global class in transaction SE24.

```
REPORT zcl_progress_demo.  
PARAMETERS:  
    status(100) LOWER CASE DEFAULT 'Progress indicator demo'(001),  
    treshold TYPE i DEFAULT 5 OBLIGATORY,  
    max TYPE i DEFAULT 10000 OBLIGATORY.  
  
DATA:  
    gauge TYPE REF TO zcl_progress_indicator.  
  
INITIALIZATION.  
    CREATE OBJECT gauge.  
  
START-OF-SELECTION.  
    CALL METHOD gauge->start  
        EXPORTING  
            i_status_text      = status  
            i_high_bound      = max  
            i_percent_treshold = treshold.  
  
    DO max TIMES.  
        CALL METHOD gauge->advance.  
        WAIT UP TO 0 SECONDS. "Just a small delay  
    ENDDO.  
  
    CALL METHOD gauge->finish.
```

Author Bio



In his pre-SAP era Sergei Korolev was a loyal Delphi programmer, developing component libraries and proprietary applications. Since 1999 he has been working as an ABAP developer. Now he is a senior ABAP consultant at IBS Company (<http://www.ibs-company.com/>) working for various customers..