Archived:FEP causing browser to crash on S60 2nd and 3rd Editions (Known Issue)

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

The article is believed to be still valid for the original topic scope.

Overview

There is a bug in S60 browser code that can cause a crash with 3rd-party FEPs. After installing a 3rd-party FEP, the browser crashes on launch, causing a KERN-EXEC 3 panic.

Description

This bug can be worked around by implementing your own MAknEditingStateIndicator that has empty methods. This will avert the browser from crashing but may have unexpected behavior with some other components, so extended testing with other editors is advised.
To implement this indicator interface you will need a header file which is not part of the SDK until S60 3rd Edition, Feature Pack 1. The header (akneditstateindicator.h), included in the S60 3rd Edition, FP1 SDK, can be used also with previous S60 platform versions that are affected by this issue.

Solution

The dummy indicator implementation would be something like this:

```
// FepIndicator.h:
#include <akneditstateindicator.h>  // for MAknEditingStateIndicator
class CFepIndicator : public CBase, public MAknEditingStateIndicator
    {
public:
        enum TIndicatorState
            {
            EStateNone = 1,
            EStateNumeric,
            EStateAlphabat
            };
public:
    static CFepIndicator* NewL();
    ~CFepIndicator();
    virtual void SetState(TIndicatorState aState);
    // From MAknEditingStateIndicator
    void SetState(TAknEditingState aState);
    CAknIndicatorContainer* IndicatorContainer();
       // ...
    };
// FepControl.cpp:
void CExampleFepControl::ConstructL()
    {
    CreateWindowL();
    SetNonFocusing();
    RWindow& window = Window();
    window.SetOrdinalPosition(0, ECoeWinPriorityFep);
    TPoint fepControlPos(0, 0);
```

```
    SetExtent(fepControlPos, TSize(0,0));
    window.SetExtent(fepControlPos, TSize(0,0));
    window.SetNonFading(ETrue);
    // Create multi-tap input engine
    iMultiTapEngine = CExampleFepMultiTapEngine::NewL(*this);
    // Create status pane indicator
    iIndicator = CFepIndicator::NewL();
    CAknEnv::Static()->SwapEditingStateIndicator(iIndicator);
    }
 // FepIndicator.cpp:
void CFepIndicator::SetState(TAknEditingState */aState*/)
    {
    }
    CAknIndicatorContainer* CFepIndicator::IndicatorContainer()
    {
    return NULL;
    }
```