

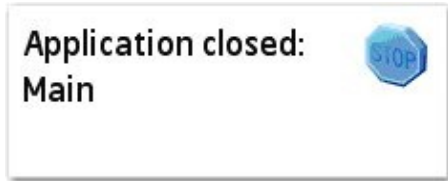
????

Contents

- 1 ??????Panic??
- 2 ??????Bug
- 3
?? UHEAP MARK? UHEAP MARKEND??????
- 4 ??????
- 5 ??????????????????
- 6 ????

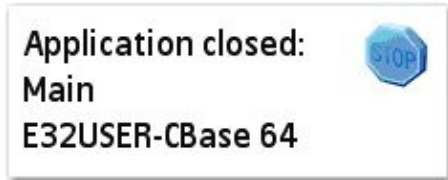
??????Panic??

?Panic?????????????????ErrRd?????????????????Panic?????????????????Panic?



?SDK

3rd????ErrRd????????????3rd?????????????????C:\Symbian\9.2\S60_3rd_FP1\Epoc32\winscw\c\resource?????????ErrRd



??? ??????3rd??SDK?ErrRd????????????????????? Tools > Preferences?????Extended panic code file?

??????Bug

??Symbian OS????????????? __ASSERT_ALWAYS
? __ASSERT_DEBUG????????????? __ASSERT_DEBUG????????????? __ASSERT_ALWAYS??

????????? __ASSERT_DEBUG??????

```
void TestValue(TInt aValue)
{
    _LIT(KPanicCategory, "TestValue");
    __ASSERT_DEBUG((aValue >= 0), User::Panic(KPanicCategory, 99));
    // Do something with aValue
    // ...
}
```

è° è¯ æ æ ¯

???????aValue??0???"Panic -99"?

?? ???????Panic? ?????????????????????????????????????Panic?????????Leave?

?????__ASSERT_DEBUG????debug?????aValue????????????????????????__ASSERT_ALWAYS?

????????????Panic????__ASSERT_DEBUG?????? ASSERT??ASSERT????????????????panic???????

???????????e32def.h???

```
#define ASSERT(x) __ASSERT_DEBUG(x, User::Invariant())
```

????????ASSERT?????

```
void TestValue(TInt aValue)
{
  ASSERT(aValue >= 0);
  // Do something with aValue
  // ...
}
```

?? __UHEAP_MARK? __UHEAP_MARKEND???????

????????????????????????????????????__UHEAP_MARK? __UHEAP_MARKEND??

```
GLDEF_C TInt E32Main()
{
  // Start checking memory leaks
  __UHEAP_MARK;

  // Create a fixed-length, flat array, which contains 10 integers
  CArrayFixFlat<TInt>* fixFlatArray;
  fixFlatArray = new(ELeave) CArrayFixFlat<TInt>(10);
  // Array is not deleted, so memory will leak

  // Stop checking memory leaks and cause a panic if there is a leak
  __UHEAP_MARKEND;

  return KErrNone;
}
```

????????????????????????????????????Panic???????



????????????????debug????????????????????????????????

???????Bug

??????

????????????? _DECLARE_TEST ?
__TEST_INVARIANT??

```

class CLivingPerson : public CBase
{
public:
    enum TGender {EMale, EFemale};
public:
    CLivingPerson(TGender aGender);
    ~CLivingPerson();
public:
    void SetAge(const TInt aAge);
private:
    TGender iGender;
    TInt iAgeInYears;
    __DECLARE_TEST; // Object invariance testing
};

CLivingPerson::CLivingPerson(TGender aGender) : iGender(aGender) {}
CLivingPerson::~CLivingPerson() {}

void CLivingPerson::SetAge(const TInt aAge)
{
    // Set age and check object invariance
    __TEST_INVARIANT;
    iAgeInYears = aAge;
    __TEST_INVARIANT;
}

void CLivingPerson::__DbgTestInvariant() const
{
    #ifdef _DEBUG // Built into debug code only
    // Person should be either male or female
    ASSERT((iGender == EMale) || (iGender == EFemale));

    // Person's age shouldn't be negative
    ASSERT(iAgeInYears >= 0);
    #endif
}

```

?????ASSERT?????????????????"USER 0"Panic?

????????????????????

```

?????????????Leave?????????????????????????????????????????????PopAndDestroy?????Pop?????????????????????????????????????CL
?
CleanupStack::PopAndDestroy?????????????????????????????????????????????????????????????????"E32USER-CBase
90" Painc????????????????????????????????????????????
CClass* obj = new(ELeave) CClass;
CleanupStack::PushL(obj);
// ...
CleanupStack::PopAndDestroy(obj); // Panics if ?obj? not on top

```

???????

è° è¯ æ æ ¯

?? ???

????

- Debugging techniques

????????????????????