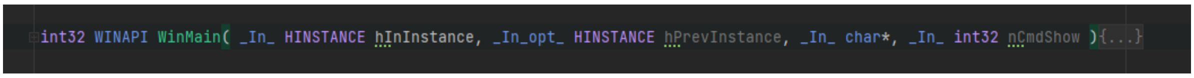
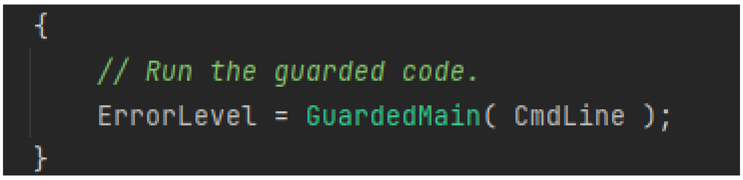
引擎启动入口：LaunchWindows.cpp



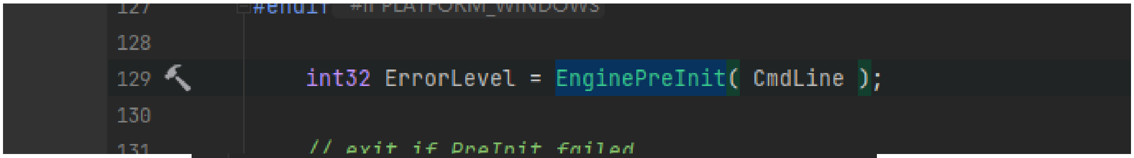
```
int32 WINAPI WinMain( _In_ HINSTANCE hInInstance, _In_opt_ HINSTANCE hPrevInstance, _In_ char*, _In_ int32 nCmdShow ){...}
```



```
{
    // Run the guarded code.
    ErrorLevel = GuardedMain( CmdLine );
}
```

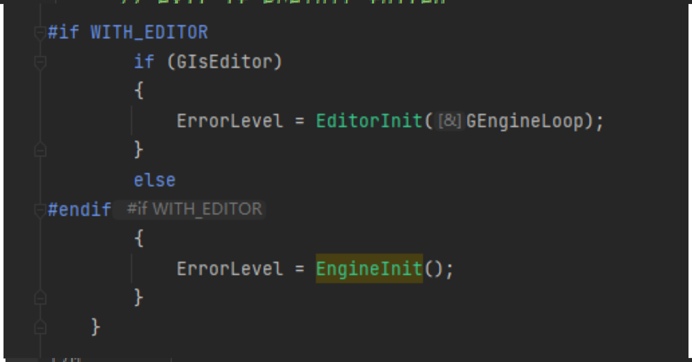
GuardMain：

1. EnginePreInit



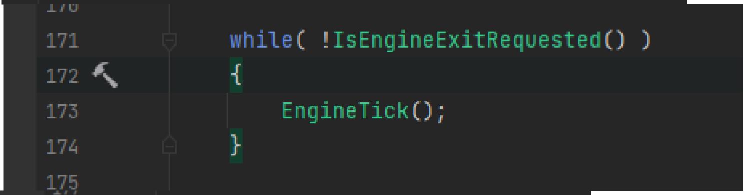
```
#endif  #if PLATFORM_WINDOWS

    int32 ErrorLevel = EnginePreInit( CmdLine );

    // exit if PreInit failed
```

2. Init



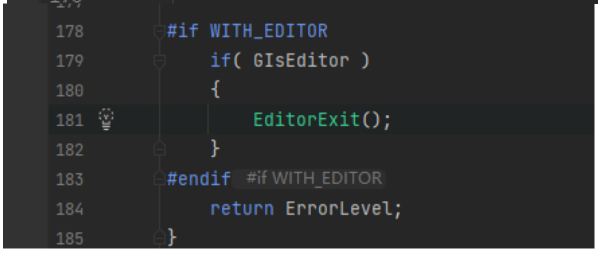
```
#if WITH_EDITOR
        if (GIsEditor)
        {
            ErrorLevel = EditorInit( [&] GEngineLoop);
        }
        else
#endif  #if WITH_EDITOR
        {
            ErrorLevel = EngineInit();
        }
    }
```

3. EngineTick



```
        while( !IsEngineExitRequested() )
        {
            EngineTick();
        }
```

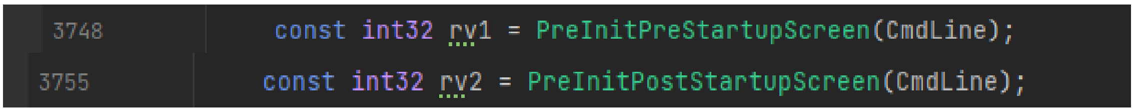
4. Exit



```
#if WITH_EDITOR
    if( GIsEditor )
    {
        EditorExit();
    }
#endif  #if WITH_EDITOR
    return ErrorLevel;
}
```

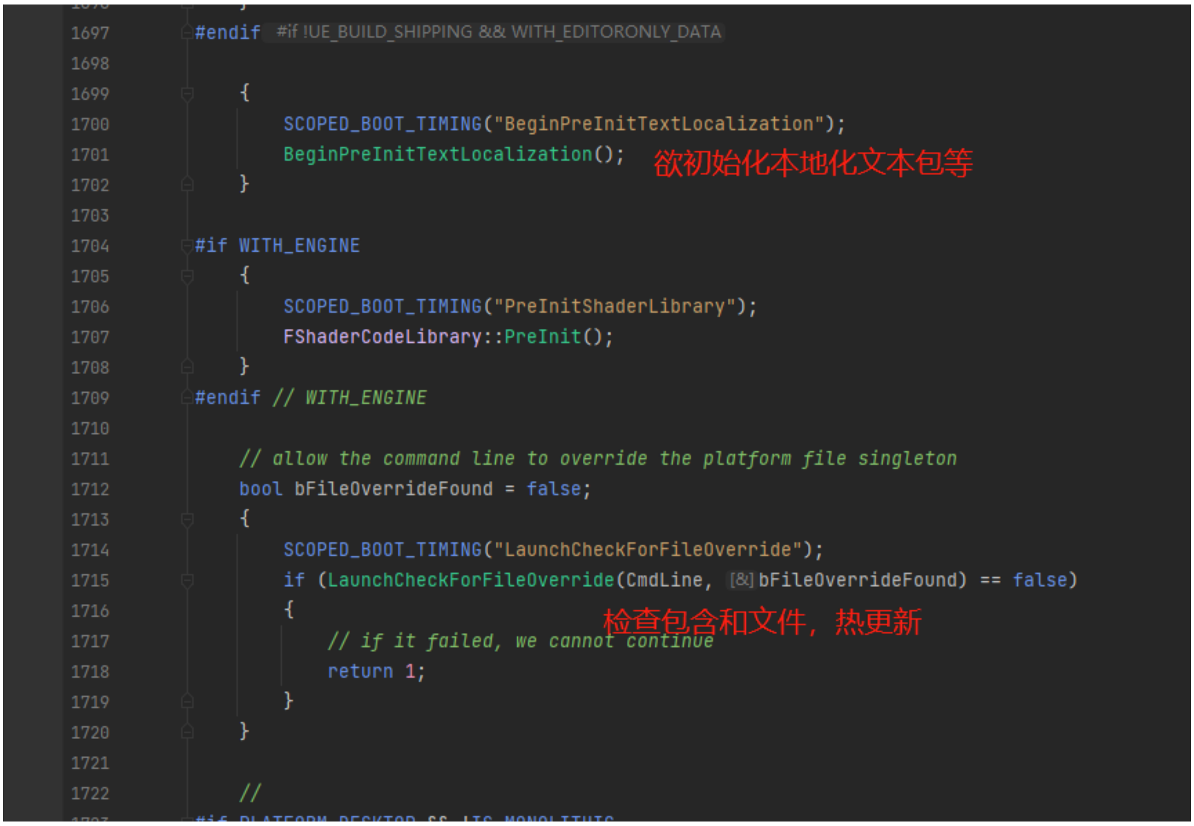
EnginePreInit

LaunchEngineLoop.cpp



1. `const int32 rv1 = PreInitPreStartupScreen(CmdLine);`
2. `const int32 rv2 = PreInitPostStartupScreen(CmdLine);`

PreInitPreStartupScreen



```
    }
#endif  #if !UE_BUILD_SHIPPING && WITH_EDITORONLY_DATA

    {
        SCOPED_BOOT_TIMING("BeginPreInitTextLocalization");
        BeginPreInitTextLocalization();
    }

#if WITH_ENGINE
    {
        SCOPED_BOOT_TIMING("PreInitShaderLibrary");
        FShaderCodeLibrary::PreInit();
    }
#endif // WITH_ENGINE

    // allow the command line to override the platform file singleton
    bool bFileOverrideFound = false;
    {
        SCOPED_BOOT_TIMING("LaunchCheckForFileOverride");
        if (LaunchCheckForFileOverride(CmdLine, [&] bFileOverrideFound) == false)
        {
            // if it failed, we cannot continue
            return 1;
        }
    }

    //
#if PLATFORM_DESKTOP && !IS_MONOLITHIC
```

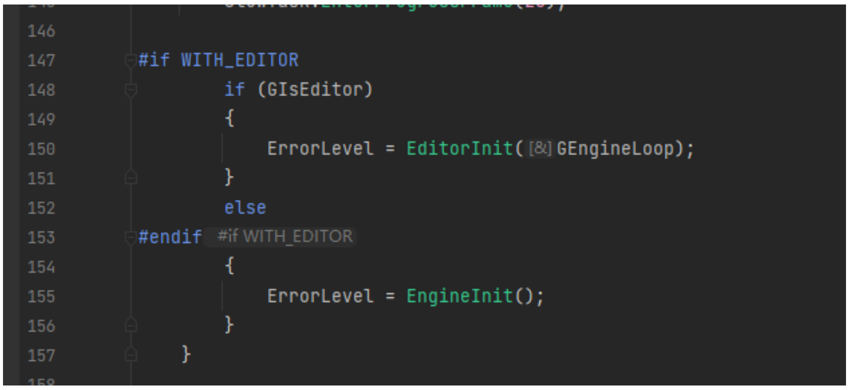
欲初始化本地化文本包等

检查包含和文件，热更新

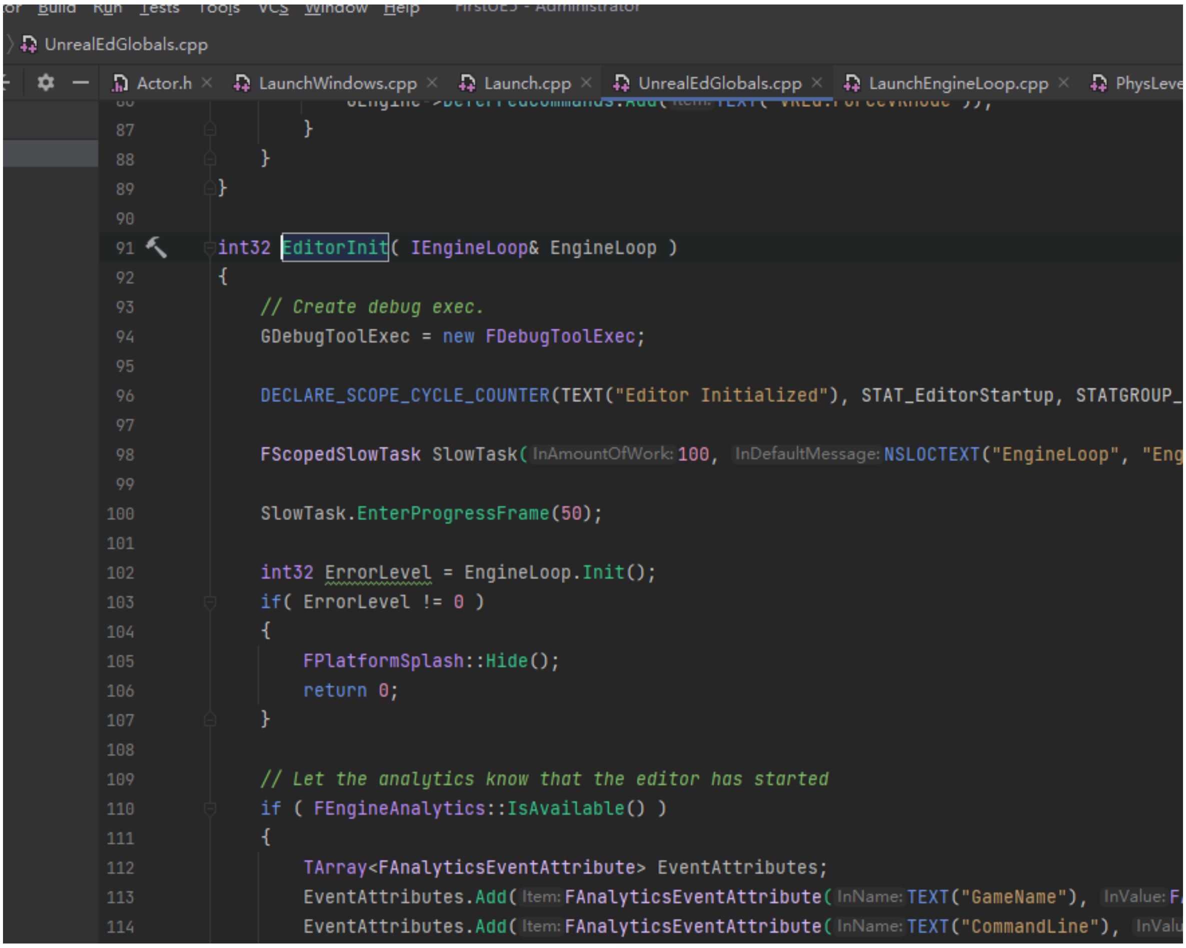
PreInitPostStartupScreen

EditorInit和EngineInit

在GuardedMain中有EditorInit和EngineInit方法，根据是否是编辑器模式来决定通过哪种方式启动。

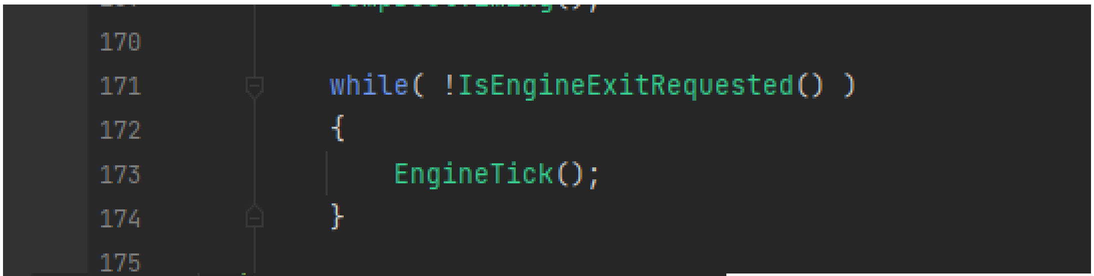


```

#if WITH_EDITOR
        if (GIsEditor)
        {
            ErrorLevel = EditorInit( [&] GEngineLoop);
        }
        else
#endif  #if WITH_EDITOR
        {
            ErrorLevel = EngineInit();
        }
    }
```

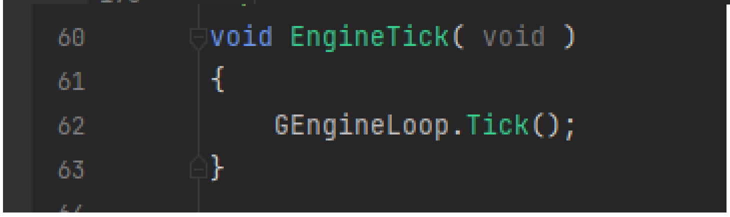


```
            GEngine->DeferredCommands.Add( Item: TEXT( "VREd.ForceVRMode" ));
        }
    }
}

int32 EditorInit( IEngineLoop& EngineLoop )
{
    // Create debug exec.
    GDebugToolExec = new FDebugToolExec;

    DECLARE_SCOPE_CYCLE_COUNTER(TEXT("Editor Initialized"), STAT_EditorStartup, STATGROUP_

    FScopedSlowTask SlowTask( InAmountOfWork: 100,  InDefaultMessage: NSLOCTEXT("EngineLoop", "Eng

    SlowTask.EnterProgressFrame(50);

    int32 ErrorLevel = EngineLoop.Init();
    if( ErrorLevel != 0 )
    {
        FPlatformSplash::Hide();
        return 0;
    }

    // Let the analytics know that the editor has started
    if ( FEngineAnalytics::IsAvailable() )
    {
        TArray<FAnalyticsEventAttribute> EventAttributes;
        EventAttributes.Add( Item: FAnalyticsEventAttribute( InName: TEXT("GameName"),  InValue: F
        EventAttributes.Add( Item: FAnalyticsEventAttribute( InName: TEXT("CommandLine"),  InValu
```

EngineTick



```

    while( !IsEngineExitRequested() )
    {
        EngineTick();
    }

```



```
void EngineTick( void )
{
    GEngineLoop.Tick();
}
```