潘荣涛

UIT编译与集成

介绍UIT编译方式，集成方式

德赛西威

[ 统一编译集成方式 3](#_Toc19201)

[ 编译 3](#_Toc25472)

[ uit-repo 3](#_Toc12635)

[ 目录结构 3](#_Toc17010)

[ WIN32 4](#_Toc7465)

[ Linux 6](#_Toc10560)

[ Uit-project 7](#_Toc7606)

[ 目录结构 7](#_Toc28117)

[ Win32 7](#_Toc19939)

[ Linux 8](#_Toc29094)

[ 集成 8](#_Toc23300)

[ 编译uit-repo对应的版本 8](#_Toc32675)

[ 同步SDK 9](#_Toc5007)

[ 基于新SDK开发 9](#_Toc9892)

[ 编译uit-project 9](#_Toc17479)

[ 对外公布UIT更新 9](#_Toc23658)

# 统一编译集成方式

UIT的编译环境将做最大的努力来达到统一，避免不同平台下不同的编译方式。简单来说，将统一工具和编译集成流程，不做平台区分。

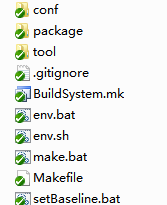
请先阅读<【uit】【框架】：工具.docx>进行编译前准备。

# 编译

## uit-repo

### 目录结构

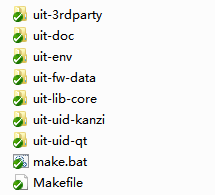
uit-repo本身也是一个数据库，它存储的是编译配置脚本，工具以及统一编译各个模块的脚本。



conf：存放各平台编译配置脚本，配合env.bat|env.sh脚本使用

tool：存放工具

package：各模块的数据库应当存放在此处。所有数据库下载完毕后大致如下



注意：目录中各模块的内容并非存放在uit-repo数据库中，而是要求模块下载放在uit-repo/package而已。

Uit-3rdparty：负责第三方库的维护

Uit-doc：所有技术框架/模块的文档存放处

Uit-env：运行时脚本，配置文件

Uit-fw-data：数据驱动模块

Uit-lib-core：基础模块，包含日志和运行时路径等

Uit-uid-kanzi：KANZI插件与显示进程模块

Uit-uid-qt：Qt 方案的封装

Uit-fw-screen：图层管理模块

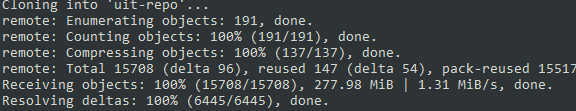
### WIN32

win32下使用cmder作为控制台工具。按照下列步骤，即可完成相应的编译。

* 下载uit-repo（假设当前目录为D:/uit）

cd d:/uit

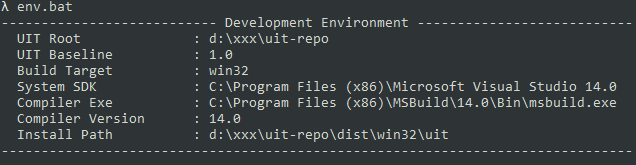
git clone git@scm-gitlab:uit/uniform/kunp/uit-repo.git



* 执行环境变量：

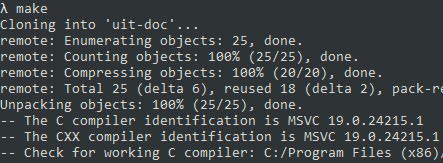
cd uit-repo

setBaseline 1.0（可选命令，不执行则默认在最新分支；但假如最新分支为2.0，你需要编译1.0则必须执行该命令）

env.bat

* 编译
* 编译所有

make：此命令将编译package下所有需要编译的数据库模块，假如该数据库未下载，将自动下载。



* 编译特定模块

cd package

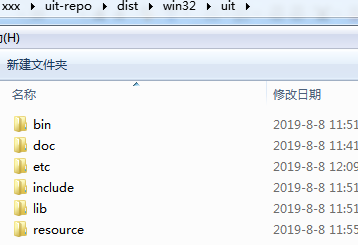
make uit-fw-data：表示编译模块uit-fw-data，它会自动编译所依赖的模块，比如它依赖uit-3rdparty，则会先编译uit-3rdparty。依赖关系请查看该数据库下的BuildSystem.Depends文件

* 清除编译

make clean

make clean\_uit-fw-data

* 查看编译输出物（uit-repo/dist下）



### Linux

* 下载system-sdk到/work/sdk下

git clone git@scm-gitlab:sdk/sv-g6sh-qnx-system-sdk.git

* 下载uit-repo

git clone git@scm-gitlab:uit/uniform/kunp/uit-repo.git

* 执行环境变量（参数为平台名，支持linux-x11, linux-arm, qnx700等）

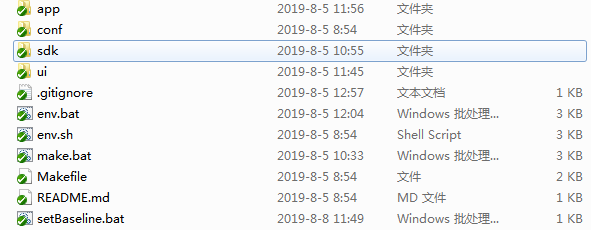
. env.sh qnx700-aarch64

* 编译

编译方式与win32环境下一致，不再赘述

## Uit-project

### 目录结构



conf：存放各平台编译配置脚本，配合env.bat|env.sh脚本使用

ui：ui相关工程的存放位置

app：app工程的存放位置

sdk：存放uit sdk的路径

### Win32

* 下载uit-project（假设当前目录为D:/uit，与uit-repo并列）

git clone git@scm-gitlab:uit/uniform/projects/uit-project.git

* 切换到项目分支

cd uit-project

setBaseline.bat g6sh

* 编译

make

* 编译特定模块

cd ui

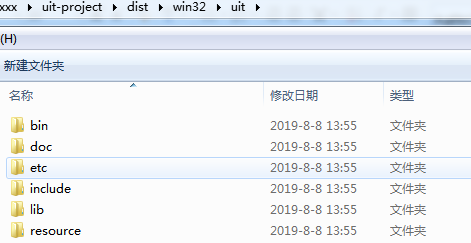
make xxx

或

cd app

make xxx

* 查看输出物（uit-project/dist）



与uit-repo的输出物很相似，只是多了UI工程和APP工程的输出。

### Linux

与win32步骤一致，不再赘述

# 集成

UIT的集成工作由UIT集成人员进行操作，集成目录就是在uit-project中进行。

下面是集成步骤。

## 编译uit-repo对应的版本

例如，假设正在进行的uit-project选用的是UIT1.0版本，则编译uit-repo的1.0版本。如果是2.0则编译2.0版本。

## 同步SDK

把编译uit-repo得到的输出物同步至uit-project下的sdk目录

## 基于新SDK开发

通知uit-project相关开发人员，在新SDK下进行开发更新

## 编译uit-project

编译整个uit-project，并上传至数据库。

## 对外公布UIT更新

对外公布该uit-project分支更新的消息。