

# Actor的创建流程

## 1. 检查是否具备创建条件

1. 待创建的类是否有效，比如没有被弃用，不是抽象类等
2. Level是否有效
3. 检查生成点的碰撞条件

## 2. 构造Actor对象

```
// actually make the actor object
AActor* const Actor = NewObject<AActor>(Outer:LevelToSpawnIn, Class, NewActorName, ActorFlags, Template,
```

## 3. 注册到Level

```
LevelToSpawnIn->Actors.Add( Actor );
LevelToSpawnIn->ActorsForGC.Add(Actor);
```

## 4. 初始化Actor对象

在初始化Actor对象组件前，会先发送一个预初始化的多播通知：

```
// Broadcast delegate before the actor and its contained components are initialized
OnActorPreSpawnInitialization.Broadcast(Actor);
```

紧接着在PostSpawnInitialize方法中初始化Actor

1. 设置Owner，Instigator（Owner是由哪个Actor生成，Instigator是由哪个Pawn生成）
2. 注册组件：在构造Actor对象时，构造了该Actor的组件。所以，在这步操作中，首先初始化根组件，然后调用RegisterAllComponents方法将所有组件进行注册即可。由此我们可以看出，组件的添加分两步：构造和注册。所以如果在Actor创建流程外要为Actor添加新的组件，需要在构造完组件后手动调用组件的注册方法才能把组件成功添加到Actor中。

## 3. 最后一步：AActor::FinishSpawning

1. 位置处理
2. ExecuteConstruction：执行蓝图构造脚本等
3. PostActorConstruction：初始化组件，执行BeginPlay等

## 5. Actor创建完成

调用OnActorSpawned通知Actor创建完成，并将Actor添加到网络同步列表。