**外部C++**

图1给出了COStream程序的编译流程，COStream编译器是一个源到源的编译器，将COStream程序根据目标平台编译成并行的目标代码（这里是C++），然后调用底层的编译器生成最终的目标代码。



图1： COStream程序的编译流程

COStream编译器支持两种方式的程序编译：独立模式和嵌入模式。以目标语言C++为例，在独立模式中，COStream程序被翻译成C++最后编译成独立可执行的代码，无需其他干涉；在嵌入模式中，COStream程序通过提供的外部接口被外部C++程序调用，成为C++的一个调用库，和外部C++一起编译连接，只需设置编译选项，编译器便能对代码以不同的方式编译：embed表示嵌入式调用，而normal表示独立执行程序。这里对嵌入模式中COStream编译器提供的接口进行简单介绍。

这里对嵌入模式中COStream编译器提供的接口进行简单介绍。在编译生成的文件中，RunCOStream.h文件定义了类RunCOStream，类RunCOStream包含在名字空间COStream中，该类主要接口用法如下：

RunCOStream(stringstream& instream);

RunCOStream(void\* pSrc,int srcDataCount);

RunCOStream(string path)；

void Run();

int GetOutputNum();

void\* GetOutputArray();

用户可以提供字符流、数组指针和数组长度以及文件路径以初始化RunCOStream，调用该类的Run方法将运行流程序，最后通过调用该类的GetOutputArray方法取得输出数据数组指针，调用GetOutputNum方法可以获得输出数据项个数；COStream.h文件包含了该类的声明，用户只需包含该文件。

对COStream的用法示例如下:

#include "RunCOStream.h"

using namespace COStream;

int main()

{

double \*p = new double[1379];

RunCOStream costream = new RunCOStream(p,1379);

costream.Run();

int retDataCount = costream.GetOutputNum();

double\* retArray = (double\*)costream. GetOutputArray();

}