

# Impulse C - Recommended Workflow

*A description of the workflow that should be followed to integrate Impulse C applications into a POSIX-compliant operating system*

---

## Contents

[Introduction](#)

[Tutorials](#)

[Templates](#)

[Tools](#)

[Example Projects](#)

## Introduction

This document describes the set of tutorials, instructions, and other miscellaneous reports provided with this project submission. It is a quick reference to new developers who want to use these materials to start integrating their Impulse C application into an POSIX-compliant real-time operating system.

## Tutorials

### Impulse C Integration Tutorial

This tutorial guides the reader through the process of creating an Impulse C application, implementing it within a PowerPC-based system for the ML507, and then running the Impulse C application from POSIX threads within the Xilkernel.

### Impulse C BubbleSort Tutorial

This tutorial is an extension of the previous one which discusses the integration process of a different Impulse C application (that is, a simple bubble sort implementation with worst-case time complexity).

## **Xilkernel Tutorial**

This tutorial guides the reader through the process of creating a PowerPC-based system that is synthesized and controlled using the Xilkernel.

## **Impulse C Instrumentation Library**

This document describes the usage of the profiling and instrumentation library we developed to aid developers who require simple cycle-based measurements of their applications.

## **Impulse C Design for RTOS Integration**

This document describes the ideal Impulse C design that is most suitable towards

## **Partial Walk-through for Partial Reconfiguration**

A partially complete document that details the steps we took when attempting to add support for partial reconfiguration. Issues encountered and our attempts to resolve them are discussed.

# **Templates**

## **Impulse C template**

This is a sample project that can be used to build new Impulse C applications that conform to the recommended interface requirements for easy POSIX integration.

## **XPS template**

This is a blank XPS project for a PowerPC-based system that has all of the components necessary to support the Xilkernel RTOS.

# **Tools**

## **Translation script (impulse\_convert.py)**

This script aids the user in converting an existing Impulse C application to POSIX-compliant software files that can be easily integrated into the Xilkernel RTOS (or any other POSIX-compliant RTOS). It parses the software files and performs automatic translation steps based on the type and interface to each software process and the usage of the profiling library.

## **Profile library (ProfileLibrary.h)**

This header files provides a set of C macros that can be used to easily gather cycle measurements for measuring the performance of an Impulse C application.

## Example Projects

- BasicStream (located in ImpulseSystem)
- HelloWorld (located in ImpulseIntegrationSystem)
- BubbleSort (located in TestSystem)
- AES (located in AESSystem)