

# DSix Project Wizard Tool

# Creating a DSix Project

## Description

This section provides an introduction to the DSix Project Wizard tool. The project wizard interface will be explored, and project configuration options will be discussed. An empty DSix project will be created and the project source will be examined. The source code will then be compiled in Visual Studio® and the project will be loaded into DSix.

## What you will learn

*Upon completion of this section, you will be able to:*

- Use the DSix Project Wizard to **create** an empty DSix project
- Identify sections of the project source code that are commonly **customized**
- Successfully **build the project** source and load the project in DSix

## Creating a DSix Project

A DSix Project is a collection of code, data, resources, settings and libraries required to run a simulation.

There are numerous approaches to generating custom DSix project, the most common being to build a blank project, created by the **Project Wizard** included with DSix *Developer* edition.

This section describes how to use the Project Wizard to create a new project and build the DLL in Visual Studio 2005 or 2008.

## Launching the Project Wizard

- ✈ Ensure the project wizard module (*DSixWizards.d6x*) is currently loaded in DSix
  
- ✈ File -> New Project

## DSix Project Wizard Options

### Development Environment

Visual Studio 2008, 2010, 2012

Visual Studio 2013, 2015, 2017

### Deployment Options

Open in Visual Studio

Load in DSix

### Project Name/Location (naming: no spaces or special characters)

# DSix Project Architecture

 Dictionary

 Data

 FlightModel

 Simulink

 TGT

## DSix Project Architecture (Dictionary)

**Dictionary** is a shell project that contains the BAR Dictionary files used to define the data used by the flight model. Many DSix tools as well as other applications use dictionary files to interpret the data in the simulation.

The **BARDictionary** application processes these files to produce buildable output for the flight model, files for network access, simulation control, Simulink models, etc.

Building this project results in BAR Dictionary being called to process each of the **.dic** files contained in the Dictionary folder.

## DSix Project Architecture (Data)

**Data** is a place holder project used to hold data files intended to be included with the development environment. These can be data tables, flight test data, etc.

The project does not actually build and produce anything, but it does support adding and deleting elements in most version control software that runs in Visual Studio.



## DSix Project Architecture (Flight Model)

**FlightModel** contains all the sources to build the DLL that DSix loads with the project.

The DSix source code is extensive, and will be discussed separately in various topics during training.

## DSix Project Architecture (Simulink)

**SIMULINK** contains the Simulink files needed to run functions created in Simulink.

## DSix Project Architecture (TGT)

**TGT** represents the output folder for all the DSix project files, Visual Studio files and DSix scripts.

Generally, this folder can be packaged separately as a run time version of the model, independent of proprietary developer information.

# Flight Model Architecture

- ✈ Flight Model Components
- ✈ Phases of Simulation
- ✈ Sample Simulation Loop

# Examples of Flight Model Components

## **Project Base** (Components that do not vary between projects)

- Timing
- State
- Environment
- Equations of Motion

## **Model Dependent** (Air-vehicle-specific components)

- Weight and Balance
- Aerodynamics
- Propulsion

## Simulation Phases

### Initialization: OnInit()

Occurs only when the flight model is first loaded

### Flight Model Reset: OnReset()

Occurs after “start” and prior to first “OnStep”

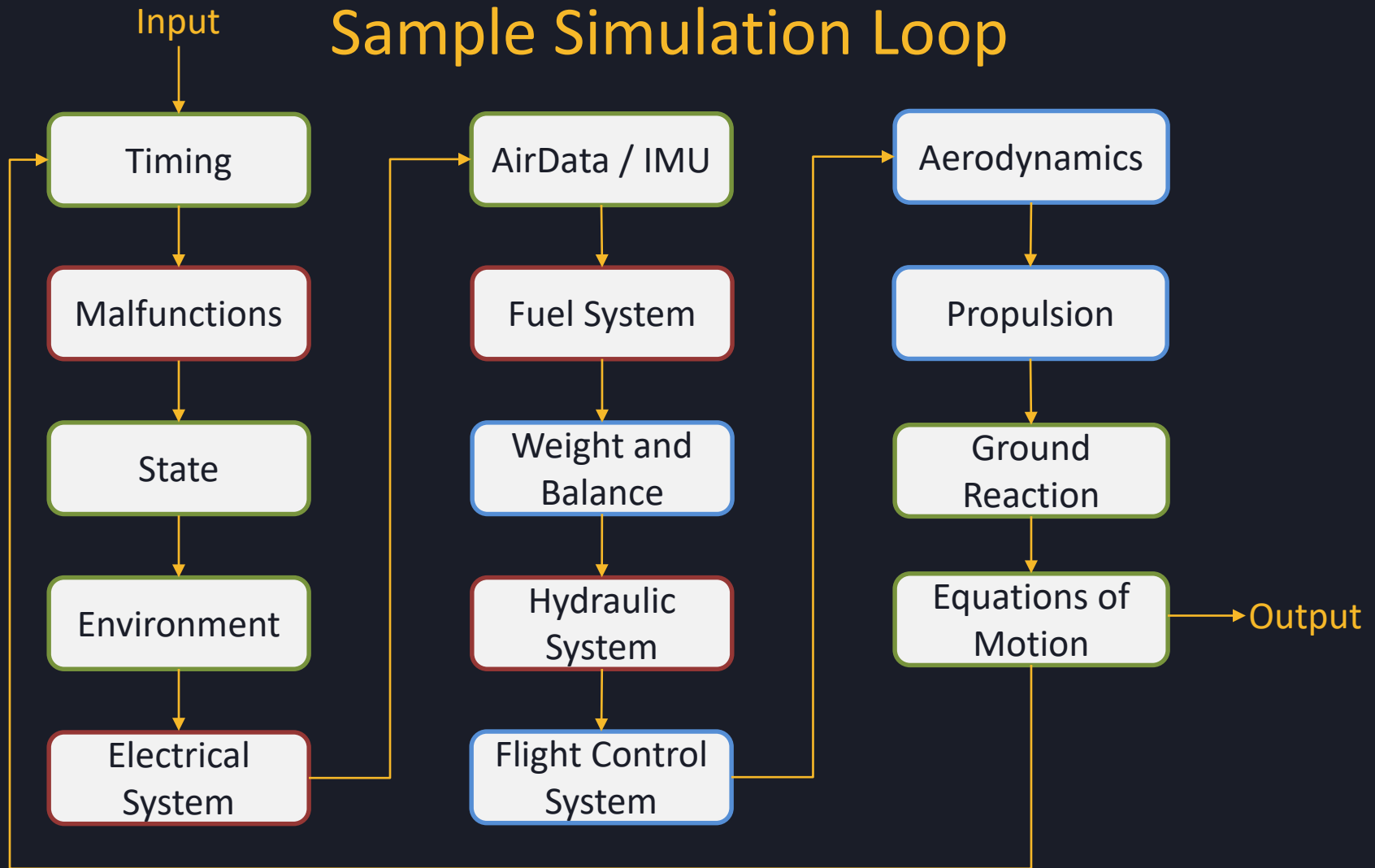
### Flight Model Step: OnStep()

The primary simulation loop during runtime operation

### Flight Model Stop: OnStop()

Occurs when a “stop” command is issued

# Sample Simulation Loop





Flight Simulation Environment