

DSix Module Management

DSix Module Management

Description

The module management section introduces the concept of DSix modules and their usage. We will describe the stock modules provided with DSix, and experiment with the loading and unloading of various modules via the DSix Module Manager tool. Module dependencies will be discussed, and the module initialization file, “Modules.ini” will be examined. Finally, we will cover troubleshooting module load failure, and discuss some common reasons that particular modules may fail to load.

What you will learn

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

- Explicitly cause particular modules to load or not load, based on project requirements
- Determine which modules have successfully loaded
- Determine potential causes of module load failure
- Describe the purpose of common default DSix modules

DSix Module Management

- ✈ What is a DSix Module
- ✈ Module Classification
- ✈ Module Loading in DSix
- ✈ Module Interactions
- ✈ Troubleshooting Module Load Failure

What is a DSix Module

A DSix module is a windows dll, providing specific functionality to the DSix application.

Several DSix Modules are required “system” modules. Others may be loaded selectively, depending upon current simulation requirements.

Note: DSix modules must be loaded at application startup.

Module Classification

- Application System Modules
- User Defined System Modules
- User Modules

Application System Modules

Application System Modules:

- Are required at all times for proper DSix program operations.
- Must be loaded before all other DSix modules.

Currently distributed Application System Modules are:

IODevices.dll	Manages DSix Input/Output devices
BARDatabase.dll	Stores and Exports DSix Runs
DSixGUI.dll	Manages DSix Graphical User Interfaces
DSixPlots.d6x	Provides Plotting Tools
DSixDataEditor.d6x	Allows Database Editing
DSixTHist.d6x	Manages Selection and Creation of Run Databases
DSixScpInit.dll	Creates and Manages Script Objects














User Defined System Modules

- Maintain load order priority over user modules
- Defined for a specific user name
- Loaded in order in which they are defined

User Modules

- Standard DSix User Modules include stock modules shipped with DSix as well as custom modules created by the user or another third party.
- DSix User modules are loaded in order of appearance in the file, 'Modules.ini', and load order may be edited with the DSix **Module Manager Tool**

Stock DSix User Modules

 DSixGraphics_I.d6x	DirectX 7 Graphics
 DSixGraphics_II.d6x	DirectX 9 Graphics
 StripChart32.d6x	Streaming Runtime Data
 D6DirectXIO.d6x	Hardware I/O
 D6IOTrigger.d6x	Custom Event Triggers
 DSixSound.d6x	Sound/Event Mapping
 DSixInstruments.d6x	Instrument Displays
 DSixSnapShot.d6x	Manage/Load Vehicle State
 VarDisp.d6x	Real Time Variable List Displays
 DSixRemote.dll	External Simulation Control
 DSixLegacyView.d6x	DSix 1.x Style HUD Window
 DSixClient.d6x (<i>Requires DSix Network</i>)	Access To External DSix Server
 DSixWizards.d6x (<i>Developer Version Only</i>)	Custom Project/Module Tool

Module Loading in DSix

Windows Registry

- User System Modules

DSix Module Manager Tool

Modules.ini

- DSix User Modules

Loading User System Modules (Windows Registry)

Purposes:

Ensure load order priority

Customize module loading on a per-user basis

Module Path:

HKEY_CURRENT_USER\Software\Bihrlle Applied Research\Simulation\D-Six 2.0\Modules

Registry Value:

Type – String

Name – *<User Defined>*

Value – *<Module Name>* or *<Module Path>* (Path required if external to DSix root)

Loading User Modules (Module Manager Tool)

...Tools..Module Manager from DSix menu

Promote Module	increase load order priority
Demote Module	decrease load order priority
Add Module	Add module name to list
Remove Module	Remove module name from list

Check to enable/disable load (Must restart DSix)

Loading User Modules (Modules.ini)

File Location	DSix root directory
File Type	Text
File Source	Module Manager Generated
Load Order	Top down

User may manually edit:

- Load Order
- Load Status
- Module names/paths (path required if module not located in DSix root directory)

Module Interactions

DSix Modules may interact with DSix or another module directly, via **registered interfaces**, or indirectly, via **global application data**.

These interactions must be considered when determining the load order of custom module(s), and the timing of code execution within the module(s).

Troubleshooting Module Load Failure

- ✈ Module Currently Loaded
- ✈ Module Not Found (Location, Spelling)
- ✈ Missing DSix Module Dependencies (Load Order)
- ✈ No License/Missing Key
- ✈ Missing External Dependencies (DirectX, Runtime Libraries)
- ✈ Deployment of Debug build



Flight Simulation Environment