Daktronics BettorData Racing Interface Manager - Overview & Setup

The **Daktronics BettorData Racing Interface Manager** is a web-based user interface that runs under the Daktronics Data Server. It consists of two web user interfaces: **Administration** and **Judges**. Information on working with the Daktronics BettorData Racing Interface Manager is covered in three quick guide documents listed below, each targeting a specific user audience.

- **Overview & Setup (DD3405451)** - This quick guide provides a functional overview and setup instructions for the Daktronics BettorData Racing Interface. This document targets Daktronics Technical Support and is not intended for non-Daktronics Administration Console operators or race judging personnel.

- **Administration Console (DD3405423)** - This quick guide provides web interface assistance to the Daktronics-trained operator. This guide covers all the necessary user controls and options.

- **Judges Console (DD3405367)** - This quick guide provides web interface assistance to the non-Daktronics trained operator. This guide covers all the necessary user controls and options required by the race judge(s).

Functional Overview

This document covers the Show Control and data system configuration steps that are necessary for the implementation and operation of the Daktronics BettorData Racing Interface Manager.

Basic Concepts

- **Meeting** - A meeting is a set of races. There is a limit of 12 races in a single meeting. On any given day there will be several meetings going on at several different venues.

- **Horses** - There can be a maximum of 24 horses in a single race; however it is not uncommon for a race to have fewer (such as only 12 horses in a single race).

- **On-Track** - On-track or on-site refers to races taking place at the track where the operator is located.

- **Off-Track** - Off-track or off-site refers to races taking place at tracks in other physical locations.

Display Presentation - Content Zones

Figure 1 provides an example of content types that may be presented in various zones of the display. The actual zone configuration will vary for each racetrack and will range from basic to more complex. Understanding the zone layout makes system setup and Show Control button creation easier and more effective.

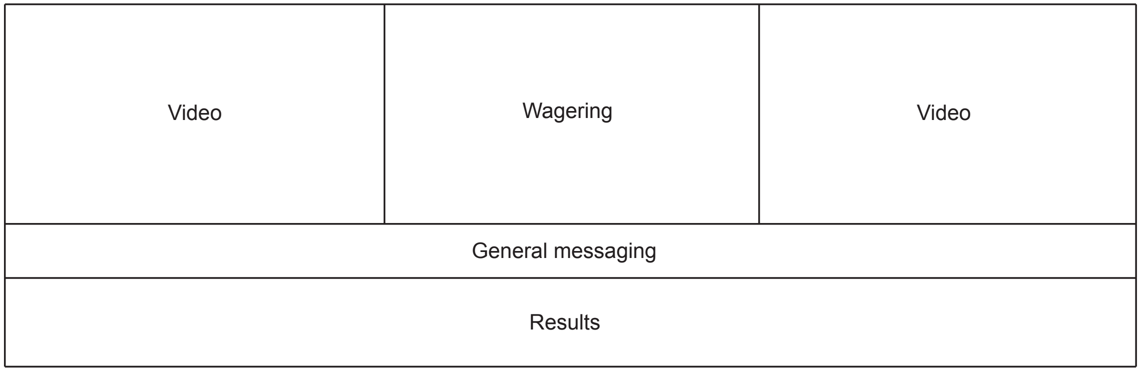


Figure 1: *Content Zones*

Although the final zone layout varies from racetrack to racetrack, all include a zone for wagering information and a zone for results information. The layout in the example in **Figure 1** also includes two video zones and a general message zone which do not rely on information generated via the BettorData Racing Interface.

- **Wagering Zone** - **Wagering** content provides information about the next race. The Show Control operator selects when to advance events in the **Wagering** zone. This is performed separately from the **Results** zone advancement.

- **Results Zone** - **Results** content provides information about the race that just completed. The Show Control operator selects when to advance events in the **Results** zone. This is performed separately from the **Wagering** zone advancement.

  Note: It is possible to update the information presented in the **Wagering** zone and the **Results** zone at the same time but this is not required. Updating the **Results** and **Wagering** zone information is determined by each racetrack location.

Theory of Operation

The following steps outline the recommended work flow order for BettorData and the Daktronics BettorData Racing Interface.

1. Set up preliminary ports. Set up the two Show Control Stats Request pre-race and post-race buttons for each active meeting and all off-track meetings that will be shown concurrently.

2. Understand the incoming data from BettorData.

    - **Output Window** - The BettorData sources provide data for four hours prior to a race. This data output is only available on the day of the race event (no data re-broadcast).

    - **Volume** - BettorData sends race data from all races, not just the races at a particular racetrack. Due to this extremely large volume of data, the data is sent in a staggered way - data for certain races is sent at certain times. Data Server must filter out the irrelevant data from this large volume of data.

    - **Relevant Data** - When the BettorData computer is launched and data begins flowing into Data Server, none of it may match the data requested for a particular race event. The BettorData system is sending data and Data Server is receiving data but the desired data may not be available. This occurs when the operator wants to pick a particular meeting from the **Administration Console** but the meeting is not listed because the data is not yet available. To give Data Server enough opportunity to gather the requested data for any particular race event, the BettorData computer should be launched as early as possible prior to the start of a meeting.

3. Power up Data Server. Data Server begins monitoring for input data from the selected primary or backup BettorData server.

4. Power up the local BettorData computer if available. BettorData outputs only to the local BettorData computer (client). This local computer must be turned on for BettorData to be available to Data Server.

    Note: Although the BettorData computer is not located in the control room and is not under Daktronics control, it is important that the local BettorData computer is powered up as soon as possible (a minimum of 45 minutes) prior to the race event and the earlier the better.

5. Request data from Data Server using the Venus® 7000 or Show Control controller. When Data Server receives a request, it begins sending out BettorData information as an RTD stream on the request port. Refer to **Requesting Data from Data Server using Control Buttons** on page 4.

6. Open the **Administration Console** and pick the desired meeting. If the meeting is not available, check back later as the data may not be available yet.

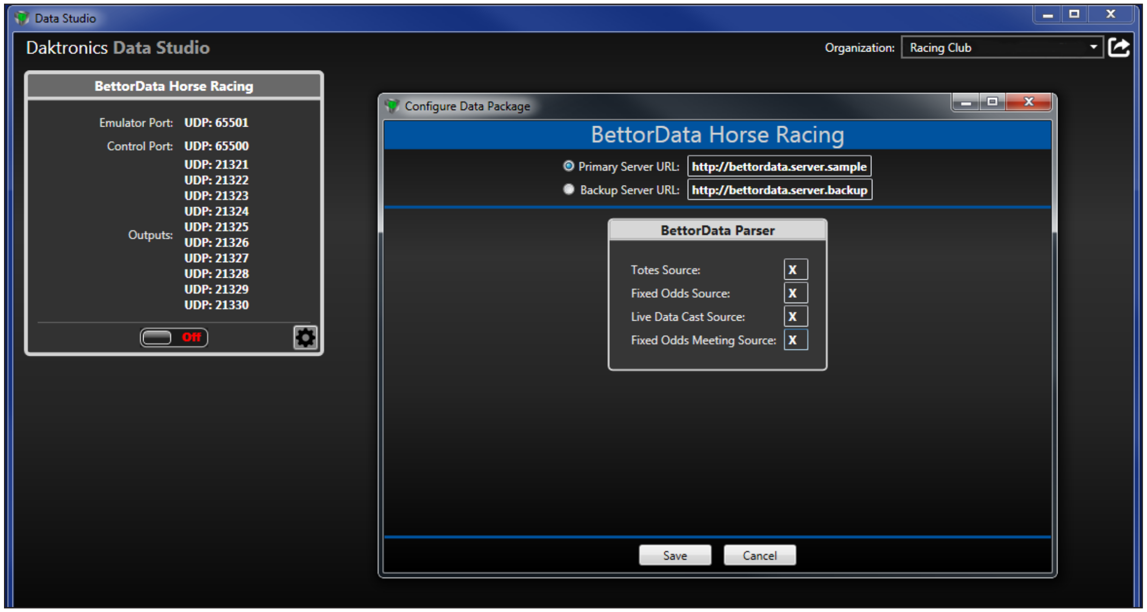
7. Open the **Judges Console**. The race judge should enter the active meeting data into the appropriate data fields within the **Judges Console**:

    - Pre-race data - **Track** and **Rail**
    - Post-race data - **Results**

Initial Data Server Configuration

Data Server must be configured to retrieve information from the BettorData feed. This configuration is normally performed by Daktronics technical support during the initial installation. Within the Data Studio application, enter the **Primary Server URL** and **Backup Server URL** under the **Configure Data Package** screen as shown in **Figure 2**. A local BettorData computer may be selected as the primary or backup server if available.

Enter the BettorData values into the fields **Totes Source**, **Fixed Odds Source**, **Live Data Cast Source**, and **Fixed Odds Meeting Source** under the **BettorData Parser** section and click **Save**.



Organization: Racing Club

Emulator Port: UDP: 65501
Control Port: UDP: 65500
Outputs: UDP: 21321
UDP: 21322
UDP: 21323
UDP: 21324
UDP: 21325
UDP: 21326
UDP: 21327
UDP: 21328
UDP: 21329
UDP: 21330

● Primary Server URL: http://bettordata.server.sample
○ Backup Server URL: http://bettordata.server.backup

Totes Source: X
Fixed Odds Source: X
Live Data Cast Source: X
Fixed Odds Meeting Source: X

***Figure 2:** Initial Data Server Configuration*

Requesting Data from Data Server using Control Buttons

Data Server must receive a request from Show Control to begin sending its BettorData feed as an RTD stream on the request port. The Show Control **Stats Requests** feature is used to create a pre-race “wagering/odds” port script command button and a post-race “results” port script command button (set of 2) for each active meeting and for all off-track meetings that will be shown concurrently. The following sub-sections present Stats Request Command button suggestions. For more detailed button creation information, refer to the **DD1834905 Show Control Installation and Setup Manual**.

Dynamic vs Static Buttons

Creating and then using Show Control Stats Request buttons for a racing event allows for streamlining operations and avoiding operator errors. There are two types of Stats Request buttons - dynamic and static (hard-coded).

- **Dynamic** - Dynamic Stats Request buttons use Show Control scripts to preconfigure most, but not all, actions necessary to send data to a specific zone on a display. After clicking a dynamic button, the operator must make an additional selection(s) such as selecting the desired race from a list. Dynamic Stats Request buttons work well in event situations where changing data presentations to the zones occurs less often.

- **Static** - Clicking a static Stats Request button immediately triggers the action (no additional user action is needed). Static Stats Request buttons work well in event situations where changing data presentations to the zones must occur often and rapidly and there is increasing potential for an operator to incorrectly select the track. By statically including all information in the button’s scripts, clicking the button immediately triggers the presentation to the zone without additional user interaction. This option requires two buttons to be created (**Wagering** and **Results**) for each race whether on-site or off-site. For example, 12 races require the creation of 24 static buttons.

Dynamic Buttons

Data Server must receive a request from Show Control to begin sending BettorData information as an RTD stream on the request port. The Show Control Stats Request feature is used to create a pre-race “wagering/odds” port script command button and a post-race “results” port script command button (set of 2) for each active meeting and for all off-track meetings that will be shown concurrently. Refer to **Figure 3**.

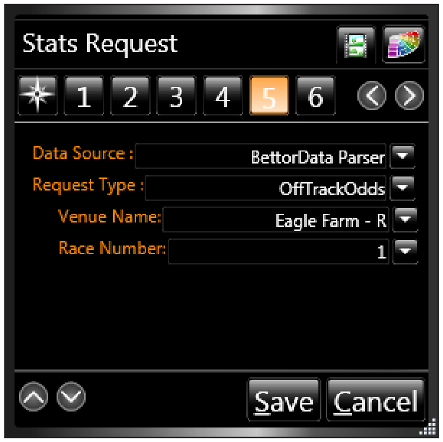
When creating a dynamic button (either **Wagering** or **Results**), enter the following values in Step 5:

- **Data Source** - Select **BettorData Parser** from the drop-down list.

- **Request Type** - Select the type of data needed from the drop-down list

- **Venue Name** - Select the appropriate venue name from the drop-down list.

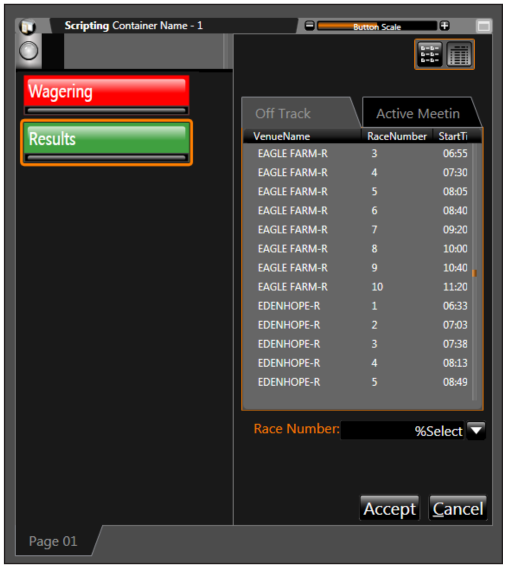
- **Race Number** - Select the correct race from the drop-down list. This information is populated by Data Server.



Data Source : BettorData Parser
Request Type : OffTrackOdds
Venue Name: Eagle Farm - R
Race Number: 1

Figure 3: Stats Request Button Creation

The scripting container shown in **Figure 4** contains unique **Wagering** (red) and **Results** (green) request buttons for a race. In this example, the **Wagering** button is active and its related stats request sidebar is shown. These **Wagering** and **Results** buttons are examples of dynamic button types.



VenueName	RaceNumber	StartTi
EAGLE FARM-R	3	06:55
EAGLE FARM-R	4	07:30
EAGLE FARM-R	5	08:05
EAGLE FARM-R	6	08:40
EAGLE FARM-R	7	09:20
EAGLE FARM-R	8	10:00
EAGLE FARM-R	9	10:40
EAGLE FARM-R	10	11:20
EDENHOPE-R	1	06:33
EDENHOPE-R	2	07:03
EDENHOPE-R	3	07:38
EDENHOPE-R	4	08:13
EDENHOPE-R	5	08:49

Race Number: %Select

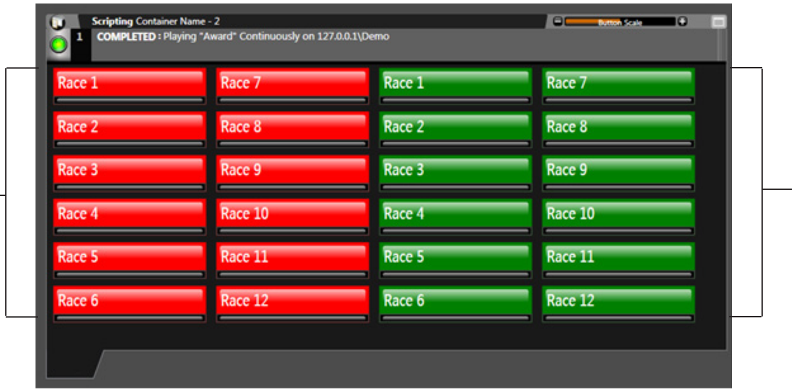
Figure 4: *Dynamic Button*

Static Buttons

Static Stats Request command buttons may be created to speed up operation. This option allows the Show Control operator to hard-code a pair of request buttons for each active on-track and off-track meeting that will be shown concurrently. A race's port request pair consists of a button for the race's **Wagering** request and a button for the race's **Results** request. The race information is hard-coded into the button so the operator is not prompted each time the button is clicked, unlike in the case of a dynamic button.

Static buttons allow a single click to activate the content with the appropriate race data in the appropriate content zone. In **Figure 5**, the race number appears on the button and the button's color indicates whether or not it is a **Wagering** port script command button (red) or a **Results** port script command button (green).

Pre-wagering buttons: button name identifies its related race; red color identifies the type as **Wagering**.



COMPLETED : Playing "Award" Continuously on 127.0.0.1\Demo

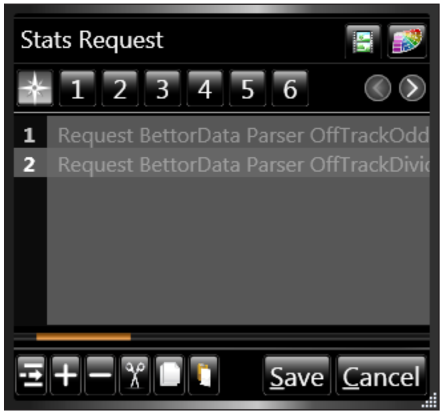
Post-results buttons: button name identifies its related race; green color identifies the type as **Results**.

This scenario has twelve races. Each race has related Stats Request Command buttons for **Wagering** (red) and for **Results** (green)

Figure 5: *Static Buttons*

Multiple Request Commands on a Single button

Show Control allows the operator to set up multiple requests in a single button. For example, a single button can activate both **Wagering** and **Results**. The **Stats Request** button in **Figure 6** shows wagering for race 3 while showing results for race 2. The Stats Request button reveals that Button Properties Step 4 has two stats requests - one for **Wagering** and the other for **Results**.



1 Request BettorData Parser OffTrackOdd
2 Request BettorData Parser OffTrackDivic

Figure 6: *Multiple Request Command*

Administration and Judges Console Startup

The BettorData Racing Interface Manager runs under Data Server and consists of two web user interfaces - **Administration Console** and **Judges Console**.

Administration Console

To start up the **Administration Console**, follow one of the options listed below. For more information refer to the **DD3405423 Daktronics BettorData Racing Interface Manager - Administration Console Quick Guide**.

- **No desktop icon** - from the designated computer in the control room, a Daktronics-trained operator opens the web browser and enters the following URL: **http://[insert the data server IP address or name here]:4455/portal**. Once this webpage opens, create a desktop icon for future use.

- **Desktop icon** - double-click the **Administration Console** desktop icon.

Judges Console

The **Judges Console** allows the race judge to perform and monitor multiple pre- and post-race tasks such as selecting a race and selecting track conditions. Only one judge can access the **Judges Console** at a time. To reach the **Judges Console**, follow one of the options listed below. For more information refer to the **DD3405367 Daktronics BettorData Racing Interface Manager - Judges Console Quick Guide**.

- **No desktop icon** - from the designated race judge location, a (preferably Daktronics-trained) judge opens the web browser and enters the URL: **http://[insert the data server IP address or name here]:4455/JudgeConsole**. Once this webpage opens, create a desktop icon for future use.

- **Desktop icon** - double-click the **Judges Console** desktop icon.

This Page Intentionally Left Blank