Greyhound Racing Victoria

GRV Data Download Centre (DDC)

Technical Guide

Document ID:	GRV - DDC - Technical Guide.docx
Version:	V1.4
Date:	05-10-2016
Status:	Final (modified)

Table of Contents

1. Document Control

1.1. Document Owner

If you have any feedback, questions, or require an electronic copy of this document, please contact:

Title: Steve Rose, General Manager Information Management and Technology
Email: srose@grv.org.au

1.2. Document Contribution List

Name	Title	Key Areas
Geoff Milner	Project Sponsor	All
Andrew Gibson	Technology & eBusiness Manager	All

1.3. Document History

Version	Date	Author	Version Update Details
0.1	Dec 2012	GM	Original Version
0.2	Apr 2013	AG	Revised with modification of XML formats
1.0	30 May 2013	AG	Finalised with Release of FastTrack
1.1	30 May 2015	AG	Alteration with inclusion of timeslot
1.2	05 October 2016	CH	Revised with form analysis for Victorian only
1.3	26 July 2017	CH	Add TimeSlot element to URL in section 5.1
1.4	OCT 2017	CH	Update contact details

1.4. Stakeholder Signoff

Name	Title	Sign-Off Received (Yes / No)	Date
Andrew Gibson	GRV Project Sponsor	Yes	

2. About this Document

2.1. Document Purpose

The objectives of this document are to:

- Describe the methods of retrieval of information from the DDC
- Describe the structure of the different file types.
- Describe the various string and decimal based data types.
- This document is designed as a Language Independent Specification (LIS).

2.2. Document structure

This document contains the following sections:

Account Details	Details of commercial account setup and testing environment
Manual Interface	Details of accessing data manually
Web Service Interface	Provides all the information required to create a valid Request to FastTrack that returns XML\CSV formatted data.
File Types	Describes the structure of the different file types of data available for download.
Data Types	Describes the structure of all of the different data types within the data files.
End to End Scenarios	Describes the several end to end data request scenarios and the expected outcome of the request.

2.3. Intended audience

This document is intended to be used by:

1. The commercial data consumer so that they can successfully interpret the data they are accessing.

3. Commercial Account

3.1. Create an Account

Setting up access to the FastTrack Data Download Centre is done via application to:

General Manager – Information Management & Technology.
Greyhound Racing Victoria

A commercial account can only be created by Greyhound Racing Victoria.

3.2. Terms & Conditions

A commercial account is created within the FastTrack system after agreement is reached on the terms and conditions within a signed Data Supply Agreement.

3.3. Security Key

Accessing an account, either manually or automatically, will require the use of a 'Security Key'.

This security key will be issued when the commercial account is established.

In the event that a commercial account needs a new security key issued, a formal request should be made to the General Manager – Information Management & Technology of Greyhound Racing Victoria.

3.4. Data Rights

Rights are granted to data based on the agreement within the terms and conditions of the Data Supply Agreement. Permissions to respective data types are established at the time an account is created.

Access to Interstate racing data (non-Victorian) will require agreement from the respective state greyhound authorities.

Rights to data can be provided for:

- Victoria
- New South Wales
- ACT
- Queensland
- South Australia
- Tasmania
- Northern Territory
- Western Australia
- New Zealand

4. Manual Download

All data provided by the FastTrack system will be available for manual download. Users that have signed a commercial agreement with Greyhound Racing Victoria can manually search for and download files they have access to.

Commercial users can also update their basic account details.

4.1. Manual Download URL

The URL to manually download files is:

https://fasttrack.grv.org.au/commercialuser/myaccount?seckey=803939c4-335b-49b0-9039-132d4c384cbd

NB: the 'seckey' in the above URL is an example only. Substitute this key for your approved security key.

4.1.1. Account Details

This page displays basic account details and a summary of the data access rights the commercial user has.

From this page the basic account details can be changed as well as gaining access to the downloads page.

4.1.2. Downloads Page

This page allows the searching of available files depending on the commercial agreement with Greyhound Racing Victoria. Files can be downloaded in either CSV or XML format.

*** PLEASE NOTE – Searches are restricted to a 31-day period at a time.

5. Web Service Interface

All data provided by the FastTrack system will be via a REST based web service. This implies that given a well formatted URL, either formatted XML or a CSV will be returned as the response to the request.

5.1. Web Service URL Structure Types

Request Meeting List	
URL Structure	https://fasttrack.grv.org.au/DataExport/<**SecurityKey**>/<**Date**>/
Description	This returns a list of the meetings held on the date supplied that the user can view.

Request Commercial Data	
URL Structure	https://fasttrack.grv.org.au/DataExport/<**SecurityKey**>/<**TimeSlot**>/<**Date**>/<**TrackCode**>/<**Report Format**>/<**FileFormat**>
Description	This is the pattern used by every commercial data request.

Request Satisfactory Trial Data	
URL Structure	https://fasttrack.grv.org.au/DataExport/<**SecurityKey**>/<**Date**>/TrialResult/<**FileFormat**>
Description	This is the pattern used by every satisfactory trial data request.

5.2. Web Service URL Elements

- ***<SecurityKey>:*** The security key assigned to a user by the network administrator for authentication.
- ***<TimeSlot>:*** The timeslot of the meeting, ie. m – morning, d – day, t – twilight and n – night.
- ***<Date>:*** The date of the meeting, formatted as dd_mm_yyyy.
- ***<TrackCode>:*** The code that identifies a particular track in accordance with national standard codes. This code has been chosen because its alignment with interstate tracks. A current list of Track Codes can be found in Section 9 of this document.
- ***<ReportFormat>:*** This determines the format of the data structure. It can be one of the following values.
  - ***"RaceResults":*** This requests the race results data format.
  - ***"Basic":*** This requests the basic field\results data format without summary analysis.
  - ***"BasicPlus":*** This requests the basic field\results data format with summary analysis.
  - ***"Full":*** This requests the full field\results data format without summary analysis.
  - ***"FullPlus":*** This requests the full field\results data format with summary analysis.
- ***<FileFormat>:*** This determines the format of the file structure. It can be either "XML" or "CSV".

5.3. Example URL

The following example URL is based on the “Request Commercial Data” structure outlined above.

http://fastrack.grv.org.au/DataExport/803939c4-335b-49b0-9039-132d4c384cbd/t/29_02_2012/300/BasicPlus/XML

NB: the ‘seckey’ in the above URL is an example only. Substitute this key for your approved security key.

6. File Types

The following is a list of the different file types that can be downloaded from FastTrack. Each provides a slightly different level of detail.

File Type Name	Description
Basic Format	This format provides just basic fields data for a meeting
Basic Plus Format	This format provides basic fields data for a meeting plus some additional form analysis data.
Full Format	This format provides the same base format as “Basic” with the addition of data for a dogs last five starts if they are available.
Full Plus Format	This format provides the same base format as “Full” with the addition of form analysis data.
Race Result Format	This format provides the results for all the races at a particular meeting.
Trial Format	This format shows the results of the satisfactory trials run at all tracks on a given date.

NB:The Basic Plus and Full Plus files contain form analysis for Victorian meetings only.

6.1. Availability of data for download

The following files are available for download once a meeting has been graded (usually 4 days before the date of the meeting):

- Basic
- Full

The following files are available for download once form analysis has been performed (usually 3-5 hours after graded meeting). The Basic Plus and Full Plus files contain form analysis for Victorian meetings only.

- Basic Plus
- Full Plus

The Race Results file is available once results have been recorded (usually 2 to 5 minutes after the completion of a race).

The Trial file is available once all trials at a meeting have been completed (usually at the end of the day)

*** PLEASE NOTE: Commercial users are able to download historical files back to the date when they signed a commercial agreement with Greyhound Racing Victoria.

6.2. Meeting List XSD Schema

The following is a sample of the XSD schema for the Meeting List request. For additional information regarding the structure of some of the string format types, refer to section “Data Interpretation” below.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Meetings">
        <xs:complexType>
```

```
                    <xs:sequence minOccurs="0" maxOccurs="unbounded">
                        <xs:element name="Meeting">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="track" type="xs:string"/>
                                    <xs:element name="date" type="xs:string"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
</xs:element>
</xs:schema>
```

6.3. Basic Format XSD Scheme

The following is a sample of the XSD scheme for the Basic Format file type. For additional information regarding the structure of some of the string format types, refer to section “Data Interpretation” below.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Meet">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Track" type="xs:string"/>
   <xs:element name="Date" type="xs:string"/>
   <xs:element name="Quali" type="xs:string"/>
  <xs:element name="Race">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="RaceNum" type="xs:integer"/>
     <xs:element name="RaceName" type="xs:string"/>
     <xs:element name="RaceTime" type="xs:string"/>
     <xs:element name="Distance" type="xs:string"/>
     <xs:element name="RaceGrade" type="xs:string"/>
     <xs:element name="PrizeMoney1" type="xs:string"/>
     <xs:element name="PrizeMoney2" type="xs:string"/>
     <xs:element name="PrizeMoney3" type="xs:string"/>
     <xs:element name="PrizeMoney4" type="xs:string"/>
     <xs:element name="PrizeMoney5" type="xs:string"/>
     <xs:element name="PrizeMoney6" type="xs:string"/>
     <xs:element name="PrizeMoney7" type="xs:string"/>
     <xs:element name="PrizeMoney8" type="xs:string"/>
     <xs:element name="GOBIS" type="xs:string"/>
     <xs:element name="Hurdle" type="xs:string"/>
     <xs:element name="Handicap" type="xs:string"/>
     <xs:element name="TAB" type="xs:string"/>
     <xs:element name="GradeCode" type="xs:string"/>
       <xs:element name="Dog">
        <xs:complexType>
         <xs:sequence>
          <xs:element name="RaceBox" type="xs:integer"/>
          <xs:element name="Last5" type="xs:string"/>
          <xs:element name="DogName" type="xs:string"/>
          <xs:element name="BestTime" type="xs:string"/>
          <xs:element name="DogHandicap" type="xs:string"/>
          <xs:element name="Trainer" type="xs:string"/>
          <xs:element name="Suburb" type="xs:string"/>
```

```
            <xs:element name="Owner" type="xs:string"/>
            <xs:element name="Sire" type="xs:string"/>
            <xs:element name="Dam" type="xs:string"/>
            <xs:element name="Colour" type="xs:string"/>
            <xs:element name="Sex" type="xs:string"/>
            <xs:element name="Whelped" type="xs:string"/>
            <xs:element name="DogGrade" type="xs:string"/>
            <xs:element name="DogGOBIS" type="xs:string"/>
           </xs:sequence>
          </xs:complexType>
         </xs:element>
       </xs:sequence>
      </xs:complexType>
     </xs:element>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
            </xs:schema>
```

6.4. Basic Format CSV Scheme

The following is a sample of the CSV scheme for the Basic Format file type. The format does not follow regular CSV principles and employs a "~" character to separate hierarchies of information. This format groups data by meeting – race. For additional information regarding the structure of some of the string format types, refer to section "Data Interpretation" below.

```
//DEV NOTES,,,,,,,,,,,,,,,,
//This scheme represents the format for a layered CSV style format of the BasicFieldScheme.xsd,,,,,,,,,,,,,,,,
//each element of a complex type are encapsulated in quotation marks and separated by a comma. the
levels of meeting and  race are separated by a tilde.,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,
<Track>,<Date>,<Quali>,,,,,,,,,,,,,,
~,,,,,,,,,,,,,,,,
<RaceNum>,<RaceName>,<RaceTime>,<Distance>,<RaceGrade>,<PrizeMoney1>,<PrizeMoney2>,<Prize
eMoney3>,<PrizeMoney4>,<PrizeMoney5>,<PrizeMoney6>,<PrizeMoney7>,<PrizeMoney8>,<GOBIS>,<H
urdle>,<Handicap>,<TAB>,<GradeCode>
<RaceBox>,<Last5>,<DogName>,<BestTime>,<DogHandicap>,<Trainer>,<Suburb>,<Owner>,<Sire>,<Da
m>,<Colour>,<Sex>,<Whelped>,<DogGrade>,<DogGOBIS>,,,
            ~,,,,,,,,,,,,,,,,
```

6.5. Basic Plus Format XSD Scheme

The following is a sample of the XSD scheme for the Basic Plus Format file type. For additional information regarding the structure of some of the string format types, refer to section “Data Interpretation” below.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Meet">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Track" type="xs:string"/>
   <xs:element name="Date" type="xs:string"/>
   <xs:element name="Quali" type="xs:string"/>
 <xs:element name="Quaddie" type="xs:string"/>
 <xs:element name="FlexQuaddie" type="xs:string"/>
 <xs:element name="Best1" type="xs:string"/>
   <xs:element name="Best2" type="xs:string"/>
 <xs:element name="Best3" type="xs:string"/>
 <xs:element name="Best4" type="xs:string"/>
  <xs:element name="Race">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="RaceNum" type="xs:integer"/>
     <xs:element name="RaceName" type="xs:string"/>
     <xs:element name="RaceTime" type="xs:string"/>
     <xs:element name="Distance" type="xs:string"/>
     <xs:element name="RaceGrade" type="xs:string"/>
     <xs:element name="PrizeMoney1" type="xs:string"/>
     <xs:element name="PrizeMoney2" type="xs:string"/>
     <xs:element name="PrizeMoney3" type="xs:string"/>
     <xs:element name="PrizeMoney4" type="xs:string"/>
     <xs:element name="PrizeMoney5" type="xs:string"/>
     <xs:element name="PrizeMoney6" type="xs:string"/>
     <xs:element name="PrizeMoney7" type="xs:string"/>
     <xs:element name="PrizeMoney8" type="xs:string"/>
     <xs:element name="GOBIS" type="xs:string"/>
     <xs:element name="Hurdle" type="xs:string"/>
     <xs:element name="Handicap" type="xs:string"/>
     <xs:element name="TAB" type="xs:string"/>
     <xs:element name="GradeCode" type="xs:string"/>
            <xs:element name="Dog">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="RaceBox" type="xs:integer"/>
         <xs:element name="Last5" type="xs:string"/>
         <xs:element name="DogName" type="xs:string"/>
         <xs:element name="BestTime" type="xs:string"/>
         <xs:element name="DogHandicap" type="xs:string"/>
         <xs:element name="Odds" type="xs:string"/>
         <xs:element name="Rating" type="xs:string"/>
         <xs:element name="Speed" type="xs:string"/>
         <xs:element name="DogComment" type="xs:string"/>
         <xs:element name="Trainer" type="xs:string"/>
         <xs:element name="Suburb" type="xs:string"/>
         <xs:element name="Owner" type="xs:string"/>
         <xs:element name="Sire" type="xs:string"/>
         <xs:element name="Dam" type="xs:string"/>
         <xs:element name="Colour" type="xs:string"/>
         <xs:element name="Sex" type="xs:string"/>
         <xs:element name="Whelped" type="xs:string"/>
         <xs:element name="DogGrade" type="xs:string"/>
         <xs:element name="DogGOBIS" type="xs:string"/>
```

```
          </xs:sequence>
         </xs:complexType>
        </xs:element>
       <xs:element name="TipsComments">
        <xs:complexType>
         <xs:sequence>
          <xs:element name="Tips" type="xs:string"/>
          <xs:element name="Leader" type="xs:string"/>
          <xs:element name="Class" type="xs:integer"/>
          <xs:element name="Wild" type="xs:integer"/>
          <xs:element name="Bet" type="xs:string"/>
          <xs:element name="FlexiFour" type="xs:string"/>
         </xs:sequence>
        </xs:complexType>
       </xs:element>
       <xs:element name="RaceComment" type="xs:string"/>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
            </xs:schema>
```

6.6. Basic Plus Format CSV Scheme

The following is a sample of the CSV scheme for the Basic Plus Format file type. The format does not follow regular CSV principles and employs a "~" character to separate hierarchies of information. This format groups data by meeting – race. For additional information regarding the structure of some of the string format types, refer to section "Data Interpretation" below.

```
//DEV NOTES,,,,,,,,,,,,,,,,,
//This scheme represents the format for a layered CSV style format of the
BasicFieldSummaryScheme.xsd,,,,,,,,,,,,,,,,,
//each element of a complex type are encapsulated in quotation marks and separated by a comma. the
levels of meeting and race are separated by a tilde.,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,
<Track>,<Date>,<Quali>,<Quaddie>,<FlexQuaddie>,<Best1>,<Best2>,<Best3>,<Best4>,,,,,,,,,
~,,,,,,,,,,,,,,,,,
<RaceNum>,<RaceName>,<RaceTime>,<Distance>,<RaceGrade>,<PrizeMoney1>,<PrizeMoney2>,<Pri
zeMoney3>,<PrizeMoney4>,<PrizeMoney5>,<PrizeMoney6>,<PrizeMoney7>,<PrizeMoney8>,<GOBIS>,
<Hurdle>,<Handicap>,<TAB>,<GradeCode>,
<RaceBox>,<Last5>,<DogName>,<BestTime>,<DogHandicap>,<Odds>,<Rating>,<Speed>,<DogComm
ent>,<Trainer>,<Suburb>,<Owner>,<Sire>,<Dam>,<Colour>,<Sex>,<Whelped>,<DogGrade>,<DogGOBI
S>
<Tips>,<Leader>,<Class>,<Wild>,<Bet>,<FlexiFour>,,,,,,,,,,,,
<RaceComment>,,,,,,,,,,,,,,,,,
            ~,,,,,,,,,,,,,,,,,
```

6.7. Full Format XSD Scheme

The following is a sample of the XSD scheme for the Full Format file type. For additional information regarding the structure of some of the string format types, refer to section “Data Interpretation” below.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Meet">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Track" type="xs:string"/>
   <xs:element name="Date" type="xs:string"/>
   <xs:element name="Quali" type="xs:string"/>
  <xs:element name="Race">
   <xs:complexType>
   <xs:sequence>
    <xs:element name="RaceNum" type="xs:integer"/>
    <xs:element name="RaceName" type="xs:string"/>
    <xs:element name="RaceTime" type="xs:string"/>
    <xs:element name="Distance" type="xs:string"/>
    <xs:element name="RaceGrade" type="xs:string"/>
    <xs:element name="PrizeMoney1" type="xs:string"/>
    <xs:element name="PrizeMoney2" type="xs:string"/>
    <xs:element name="PrizeMoney3" type="xs:string"/>
    <xs:element name="PrizeMoney4" type="xs:string"/>
    <xs:element name="PrizeMoney5" type="xs:string"/>
    <xs:element name="PrizeMoney6" type="xs:string"/>
    <xs:element name="PrizeMoney7" type="xs:string"/>
    <xs:element name="PrizeMoney8" type="xs:string"/>
    <xs:element name="GOBIS" type="xs:string"/>
    <xs:element name="Hurdle" type="xs:string"/>
    <xs:element name="Handicap" type="xs:string"/>
    <xs:element name="TAB" type="xs:string"/>
    <xs:element name="GradeCode" type="xs:string"/>
      <xs:element name="Dog">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="RaceBox" type="xs:integer"/>
         <xs:element name="DogName" type="xs:string"/>
         <xs:element name="BestTime" type="xs:string"/>
         <xs:element name="DogHandicap" type="xs:string"/>
         <xs:element name="StartsTOT" type="xs:string"/>
         <xs:element name="StartsTTD" type="xs:string"/>
         <xs:element name="Trainer" type="xs:string"/>
         <xs:element name="Suburb" type="xs:string"/>
         <xs:element name="Owner" type="xs:string"/>
         <xs:element name="Sire" type="xs:string"/>
         <xs:element name="Dam" type="xs:string"/>
         <xs:element name="Colour" type="xs:string"/>
         <xs:element name="Sex" type="xs:string"/>
         <xs:element name="Whelped" type="xs:string"/>
         <xs:element name="DogGrade" type="xs:string"/>
         <xs:element name="DogGOBIS" type="xs:string"/>
         <xs:element name="DogPRIZE" type="xs:integer"/>
          <xs:element name="Form">
           <xs:complexType>
            <xs:sequence>
             <xs:element name="Place" type="xs:string"/>
             <xs:element name="FormBox" type="xs:integer"/>
             <xs:element name="Weight" type="xs:decimal"/>
             <xs:element name="TrackCode" type="xs:string"/>
```

```
                <xs:element name="FormDistance" type="xs:string"/>
                <xs:element name="FormDate" type="xs:string"/>
                <xs:element name="FormGrade" type="xs:string"/>
                <xs:element name="FormHandicap" type="xs:string"/>
                <xs:element name="Margin" type="xs:string"/>
                <xs:element name="FirstSecond" type="xs:string"/>
                <xs:element name="DogTime" type="xs:decimal"/>
                <xs:element name="WinTime" type="xs:decimal"/>
                <xs:element name="BON" type="xs:decimal"/>
                <xs:element name="StartPrice" type="xs:string"/>
                <xs:element name="PIR" type="xs:string"/>
                <xs:element name="Split" type="xs:decimal"/>
               </xs:sequence>
              </xs:complexType>
             </xs:element>
           </xs:sequence>
          </xs:complexType>
         </xs:element>
       </xs:sequence>
      </xs:complexType>
     </xs:element>
     </xs:sequence>
   </xs:complexType>
</xs:element>
</xs:schema>
```

6.8. Full Format CSV Scheme

The following is a sample of the CSV scheme for the Full Format file type. The format does not follow regular CSV principles and employs a "~" character to separate hierarchies of information. This format groups data by meeting – race. For additional information regarding the structure of some of the string format types, refer to section "Data Interpretation" below.

```
//DEV NOTES,,,,,,,,,,,,,,,,,,
//This scheme represents the format for a layered CSV style format of the FullFieldScheme.xsd,,,,,,,,,,,,,,,,,,
//each element of a complex type are encapsulated in quotation marks and separated by a comma. the
levels of meeting  and race are separated by a tilde.,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,
<Track>,<Date>,<Quali>,,,,,,,,,,,,,,,,
~,,,,,,,,,,,,,,,,,,
<RaceNum>,<RaceName>,<RaceTime>,<Distance>,<RaceGrade>,<PrizeMoney1>,<PrizeMoney2>,<Prize
eMoney3>,<PrizeMoney4>,<PrizeMoney5>,<PrizeMoney6>,<PrizeMoney7>,<PrizeMoney8>,<GOBIS>,<H
urdle>,<Handicap>,<TAB>,<GradeCode>,,
<RaceBox>,<DogName>,<BestTime>,<DogHandicap>,<StartsTOT>,<StartsTTD>,<Trainer>,<Suburb>,<O
wner>,<Sire>,<Dam>,<Colour>,<Sex>,<Whelped>,<DogGrade>,<DogGOBIS>,<DogPRIZE>,,,
<Place>,<FormBox>,<Weight>,"<TrackCode>""",<FormDistance>,<FormDate>,<FormGrade>,<FormHandi
cap>,<Margin>,<FirstSecond>,<DogTime>,<WinTime>,<BON>,<StartPrice>,<PIR>,<Split>,<RaceGrade>,
<Checks>,<Vet>,<Jump>
            ~,,,,,,,,,,,,,,,,,,
```

6.9. Full Plus Format XSD Scheme

The following is a sample of the XSD scheme for the Full Plus Format file type. For additional information regarding the structure of some of the string format types, refer to section "Data Interpretation" below.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Meet">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Track" type="xs:string"/>
      <xs:element name="Date" type="xs:string"/>
      <xs:element name="Quali" type="xs:string"/>
   <xs:element name="Quaddie" type="xs:string"/>
      <xs:element name="FlexQuaddie" type="xs:string"/>
      <xs:element name="Best1" type="xs:string"/>
      <xs:element name="Best2" type="xs:string"/>
      <xs:element name="Best3" type="xs:string"/>
      <xs:element name="Best4" type="xs:string"/>
    <xs:element name="Race">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="RaceNum" type="xs:integer"/>
       <xs:element name="RaceName" type="xs:string"/>
       <xs:element name="RaceTime" type="xs:string"/>
       <xs:element name="Distance" type="xs:string"/>
       <xs:element name="RaceGrade" type="xs:string"/>
       <xs:element name="PrizeMoney1" type="xs:string"/>
       <xs:element name="PrizeMoney2" type="xs:string"/>
       <xs:element name="PrizeMoney3" type="xs:string"/>
       <xs:element name="PrizeMoney4" type="xs:string"/>
```

```
<xs:element name="PrizeMoney5" type="xs:string"/>
<xs:element name="PrizeMoney6" type="xs:string"/>
<xs:element name="PrizeMoney7" type="xs:string"/>
<xs:element name="PrizeMoney8" type="xs:string"/>
<xs:element name="GOBIS" type="xs:string"/>
<xs:element name="Hurdle" type="xs:string"/>
<xs:element name="Handicap" type="xs:string"/>
<xs:element name="TAB" type="xs:string"/>
<xs:element name="GradeCode" type="xs:string"/>
  <xs:element name="Dog">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="RaceBox" type="xs:integer"/>
     <xs:element name="DogName" type="xs:string"/>
     <xs:element name="BestTime" type="xs:string"/>
     <xs:element name="DogHandicap" type="xs:string"/>
     <xs:element name="Odds" type="xs:string"/>
     <xs:element name="Rating" type="xs:string"/>
     <xs:element name="Speed" type="xs:string"/>
     <xs:element name="DogComment" type="xs:string"/>
     <xs:element name="StartsTOT" type="xs:string"/>
     <xs:element name="StartsTTD" type="xs:string"/>
     <xs:element name="Trainer" type="xs:string"/>
     <xs:element name="Suburb" type="xs:string"/>
     <xs:element name="Owner" type="xs:string"/>
     <xs:element name="Sire" type="xs:string"/>
     <xs:element name="Dam" type="xs:string"/>
     <xs:element name="Colour" type="xs:string"/>
     <xs:element name="Sex" type="xs:string"/>
     <xs:element name="Whelped" type="xs:string"/>
     <xs:element name="DogGrade" type="xs:string"/>
     <xs:element name="DogGOBIS" type="xs:string"/>
     <xs:element name="DogPRIZE" type="xs:integer"/>
      <xs:element name="Form">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="Place" type="xs:string"/>
         <xs:element name="FormBox" type="xs:integer"/>
         <xs:element name="Weight" type="xs:decimal"/>
         <xs:element name="TrackCode" type="xs:string"/>
         <xs:element name="FormDistance" type="xs:string"/>
         <xs:element name="FormDate" type="xs:string"/>
         <xs:element name="FormGrade" type="xs:string"/>
         <xs:element name="FormHandicap" type="xs:string"/>
         <xs:element name="Margin" type="xs:string"/>
         <xs:element name="FirstSecond" type="xs:string"/>
         <xs:element name="DogTime" type="xs:decimal"/>
         <xs:element name="WinTime" type="xs:decimal"/>
         <xs:element name="BON" type="xs:decimal"/>
         <xs:element name="StartPrice" type="xs:string"/>
         <xs:element name="PIR" type="xs:string"/>
         <xs:element name="Split" type="xs:decimal"/>
        </xs:sequence>
       </xs:complexType>
      </xs:element>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
<xs:element name="TipsComments">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Tips" type="xs:string"/>
   <xs:element name="Leader" type="xs:string"/>
   <xs:element name="Class" type="xs:integer"/>
```

```
        <xs:element name="Wild" type="xs:integer"/>
        <xs:element name="Bet" type="xs:string"/>
        <xs:element name="FlexiFour" type="xs:string"/>
       </xs:sequence>
      </xs:complexType>
     </xs:element>
     <xs:element name="RaceComment" type="xs:string"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:element>
</xs:schema>
```

6.10. Full Plus Format CSV Scheme

The following is a sample of the CSV scheme for the Full Plus Format file type. The format does not follow regular CSV principles and employs a "~" character to separate hierarchies of information. This format groups data by meeting – race. For additional information regarding the structure of some of the string format types, refer to section "Data Interpretation" below.

```
<Track>,<Date>,<Quali>,<Quaddie>,<FlexQuaddie>,<Best1>,<Best2>,<Best3>,<Best4>,,,,,,,,,,,,
~,,,,,,,,,,,,,,,,,,,,
<RaceNum>,<RaceName>,<RaceTime>,<Distance>,<RaceGrade>,<PrizeMoney1>,<PrizeMoney2>,<Prize
Money3>,<PrizeMoney4>,<PrizeMoney5>,<PrizeMoney6>,<PrizeMoney7>,<PrizeMoney8>,<GOBIS>,<H
urdle>,<Handicap>,<TAB>,<GradeCode>,,,,
<RaceBox>,<DogName>,<BestTime>,<DogHandicap>,<Odds>,<Rating>,<Speed>,<DogComment>,<Star
tsTOT>,<StartsTTD>,<Trainer>,<Suburb>,<Owner>,<Sire>,<Dam>,<Colour>,<Sex>,<Whelped>,<DogGra
de>,<DogGOBIS>,<DogPRIZE>,
<Place>,<FormBox>,<Weight>,<TrackCode>,<FormDistance>,<FormDate>,<FormGrade>,<FormHandica
p>,<Margin>,<FirstSecond>,<DogTime>,<WinTime>,<BON>,<StartPrice>,<PIR>,<Split>,<RaceGrade>,<C
hecks>,<Vet>,<Running>,<Not Used>,<RaceNumber>
<Tips>,<Leader>,<Class>,<Wild>,<Bet>,<FlexiFour>,,,,,,,,,,,,,,,
<RaceComment>,,,,,,,,,,,,,,,,,,,,
~,,,,,,,,,,,,,,,,,,,,
```

6.11. Result Format XSD Scheme

The following is a sample of the XSD scheme for the Result Format file type. For additional information regarding the structure of some of the string format types, refer to section “Data Interpretation” below.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Meet">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Track" type="xs:string"/>
   <xs:element name="Date" type="xs:string"/>
  <xs:element name="Race">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="RaceNum" type="xs:integer"/>
     <xs:element name="RaceName" type="xs:string"/>
     <xs:element name="RaceTime" type="xs:string"/>
     <xs:element name="Distance" type="xs:string"/>
     <xs:element name="RaceGrade" type="xs:string"/>
      <xs:element name="Dog">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="Place" type="xs:integer"/>
         <xs:element name="DogName" type="xs:string"/>
         <xs:element name="Trainer" type="xs:string"/>
         <xs:element name="Box" type="xs:integer"/>
         <xs:element name="Rug" type="xs:integer"/>
         <xs:element name="Weight" type="xs:decimal"/>
         <xs:element name="StartPrice" type="xs:string"/>
         <xs:element name="Handicap" type="xs:string"/>
         <xs:element name="Margin1" type="xs:decimal"/>
         <xs:element name="Margin2" type="xs:decimal"/>
         <xs:element name="PIR" type="xs:string"/>
         <xs:element name="SplitTime" type="xs:decimal"/>
         <xs:element name="RunTime" type="xs:decimal"/>
        </xs:sequence>
       </xs:complexType>
      </xs:element>
    <xs:element name="Times">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="WinTime" type="xs:integer"/>
       <xs:element name="Split1Time" type="xs:decimal"/>
       <xs:element name="Split1Rug" type="xs:integer"/>
       <xs:element name="Split2Time" type="xs:decimal"/>
       <xs:element name="Split2Rug" type="xs:integer"/>
       <xs:element name="Split3Time" type="xs:decimal"/>
       <xs:element name="Split3Rug" type="xs:integer"/>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
    <xs:element name="Dividends">
     <xs:complexType>
      <xs:sequence>
```

```
          <xs:element name="WinDiv" type="xs:integer"/>
          <xs:element name="Place1Div" type="xs:decimal"/>
          <xs:element name="Place2Div" type="xs:integer"/>
          <xs:element name="Place3Div" type="xs:decimal"/>
         </xs:sequence>
        </xs:complexType>
       </xs:element>
       <xs:element name="Exotics">
        <xs:complexType>
         <xs:sequence>
          <xs:element name="Quin" type="xs:decimal"/>
          <xs:element name="Exacta" type="xs:decimal"/>
          <xs:element name="Trifecta" type="xs:decimal"/>
          <xs:element name="Pick4" type="xs:decimal"/>
          <xs:element name="RD" type="xs:decimal"/>
          <xs:element name="DD" type="xs:decimal"/>
          <xs:element name="Quad" type="xs:decimal"/>
         </xs:sequence>
        </xs:complexType>
       </xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

6.12. Result Format CSV Scheme

The following is a sample of the CSV scheme for the Result Format file type. The format does not follow regular CSV principles and employs a “~” character to separate hierarchies of information. This format groups data by meeting – race – track. For additional information regarding the structure of some of the string format types, refer to section “Data Interpretation” below.

```
//DEV NOTES,,,,,,,,,,,
//This scheme represents the format for a layered CSV style format of the ResultFormat.xsd,,,,,,,,,,,
//each element of a complex type are encapsulated in quotation marks and separated by a comma. the
levels of meeting, race and dog are separated by a tilde.,,,,,,,,,,,
,,,,,,,,,,,
,,,,,,,,,,,
<Track>,<Date>,,,,,,,,,,
~,,,,,,,,,,,
<RaceNum>,<RaceName>,<RaceTime>,<Distance>,<RaceGrade>,,,,,,,
<Place>,<DogName>,<Trainer>,<Box>,<Rug>,<Weight>,<StartPrice>,<Handicap>,<Margin1>,<Margin2>
,<PIR>,<SplitTime>,<RunTime>
<WinTime>,<Split1Time>,<Split1Rug>,<Split2Time>,<Split2Rug>,<Split3Time>,<Split3Rug>,,,,,,
<WinDiv>,<Place1Div>,<Place2Div>,<Place3Div>,,,,,,,,,
<Quin>,<Exacta>,<Trifecta>,<Pick4>,<RD>,<DD>,<Quad>,,,,,,
~,,,,,,,,,,,
```

6.13. Trial Format XSD Scheme

The following is a sample of the XSD scheme for the Trial Format file type. For additional information regarding the structure of some of the string format types, refer to section “Data Interpretation” below.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="TrialResultFormat">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="MeetingCode" type="xs:string"/>
      <xs:element name="DogName" type="xs:string"/>
      <xs:element name="WinTime" type="xs:decimal"/>
      <xs:element name="Place" type="xs:integer"/>
      <xs:element name="Margin" type="xs:decimal"/>
      <xs:element name="FastOnNight" type="xs:decimal"/>
      <xs:element name="Distance" type="xs:integer"/>
      <xs:element name="Result" type="xs:string"/>
      <xs:element name="TrialReason" type="xs:string"/>
      <xs:element name="DogGrade" type="xs:integer"/>
      <xs:element name="Box" type="xs:integer"/>
      <xs:element name="Weight" type="xs:decimal"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

6.14. Trial Format CSV Scheme

The following is a sample of the CSV scheme for the Trial Format file type. For additional information regarding the structure of some of the string format types, refer to section “Data Interpretation” below.

```
"<MeetingCode>","<DogName>","<WinTime>","<Place>","<Margin>","<FastOnNight>",
"<Distance>","<Result>","<TrialReason>","<DogGrade>","<Box>","<Weight>"
```

7. Data Types

The following table outlines all the different data types used in the various file types and how the different string element structures are to be interpreted. This list does not include the elements returned in a Meeting List request.

Data Type Name	Description	Required
Track	This is the full track name displayed in the FastTrack system.	Yes
Date	This is a date type that will be recorded as dd MM yy	Yes
Quali	Empty = Not a Qualifying Meeting "Q" = Qualifying Meeting	No
Status	"Unofficial = Race Results have not been finalised. "Official" = race results are finalised.	Yes
FlexQuaddie	This string is made up of 4 sets of box numbers separated by "\", with between 1 to 8 box numbers separated by a comma. The string is then terminated with the estimated winning percentage in brackets. The letter "F" can be substituted for the set of all boxes in a field and will appear on its own for a given set. E.g. 2,3,4,5,7/4,5,6,7,8/1,3,4,8/F (16.67%)	Yes
Best*N*	This is a description for a best of the day bet and will be made up of "WIN - <DogName>(R<RaceNumber>B<BoxNumber>). Maximum of 4 values	No
RaceNum	This is an integer value > 0	Yes
RaceName	This is the full race name displayed in the FastTrack system.	Yes
RaceTime	This is a time value in the 12 hour format of HH:MM<AM|PM>. Time is recorded as the local time the race is/was run E.g. 11:09AM	Yes
Distance	This is an integer value > 0 with the letter "m" appended at the end to represent "metres"	Yes
RaceGrade	This is a string value of the races grade as displayed in the FastTrack system	Yes
PrizeMoney*N*	This is an integer value > 0 with a "$" character prefix. Maximum of 8 values	No
GOBIS	This is an integer value > 0 with a "$" character prefix.	No
Hurdle	Empty = Not a Hurdle Race "Y" = Hurdle Race	No
Handicap	Empty = Not a Handicapped Race "Y" = Handicapped Race	No
TAB	A set of exotic bet types that apply to a race, separated by "\". QUAD = Quadrella TRI = Trifecta QUIN = Quinella R/D = Running Double EXACTA = Exacta	No
GradeCode	This is the abbreviation of the RaceCode. Maximum of 4 characters.	Yes
Tips	Four box numbers separated by a comma to represent a tip for the first four places.	No
Leader	0 to 8 set of box numbers representing the dogs declared as fast starters in the race. The box numbers are separated by a comma.	No

Data Type Name	Description	Required
Class	0 to 8 set of box numbers representing the dogs that have been declared as the classiest dogs in the field. The box numbers are separated by a comma.	No
Wild	0 to 8 set of box numbers representing the dogs that have been declared as possibly staging an upset the race. The box numbers are separated by a comma.	No
Bet	A complex string that represents a suggested bet for the race. Free text value.	No
FlexiFour	A complex string representing a suggested $10 Flexi Four bet. This is just open text.	No
RaceComments	Free text value.	No
RaceBox\Box	Integer value between 1 and 8.	Yes
Dog Name	String value representing the name of the dog.	Yes
BestTime	This can be a decimal value to 2 decimal places, or a code value indicating that there is no best time yet “NBT” = “No Best Time” “FSH” = “First Start Here” “FSTD” = “First Start Track and Distance”	No
DogHandicap	This is an integer value > 0 with the character “m” as the suffix to represent “metres”.	No
Trainer	A string value representing the full name of the person declared as a dog’s trainer.	Yes
Owner	A string value representing 1 of the following: Single Owner: The first initial and the surname of the owner separated by a space. Syndicate: The name of the syndicate and the first initial and surname of each member of the syndicate. The syndicate names and owners names must be separated by a comma.	Yes
Suburb	A string value representing the name of the suburb that the trainer trains from.	Yes
Sire\Dam	A string value representing the name of the dogs Sire\Dam.	Yes
Colour	A string value representing the colour of the dog.	Yes
Sex	A Character representing the sex of the dog “D” = Dog “B” = Bitch	Yes
Whelped	This is a date type that will be recorded as dd-MMM-yyyy	Yes
DogGrade	This is a single character value representing the grade of a dog. This value can be a number between 1 and 5 or the character “M” = Maiden	Yes
DogGOBIS	Empty = Not a GOBIS dog “Y” = GOBIS dog.	No
DogPrize	This is an integer value > 0 that represents the total amount of prize money that the dog has won to date.	No
Rug	This is an integer value between 1 and 10	No
Place	This is a string value from “1st” to “8th”. Where a dog fails to finish the place value can be: T = Tailed Off B = Stayed in Box F = Fell P = Pulled Up	No
FormBox	This is an integer between the 1 and 8.	No
Weight	This is a decimal value to 1 decimal place	No
TrackCode	This is a string code value that indentifies a particular track.	No
FormDistance	This is an integer value > 0	No
FormDate	This is a date type that will be recorded as dd MMM yy.	No

Data Type Name	Description	Required
FormGrade	This is a string value that represents the grade of the race a dog has raced in.	No
FormHandicap	This is an integer value that determines a dogs handicap in a race	No
Margin	Refer to the section “Margin Conversions”	No
FirstSecond	This is a string value that represents a dog’s name and finish time	No
DogTime	This is a decimal value to 2 decimal places	No
WinTime	This is a decimal value to 2 decimal places	No
BON	This is a decimal value to 2 decimal places that indicates the best time run on the night over the races distance and grade	No
StartPrice	This is an integer value > 0 that is prefixed by the character “$”.	No
PIR	This is a dogs place at each of the split points in a race. The string representation is <Speed>/<PlaceAtSplit1><PlaceAtSplint2> <PlaceAtSplitN> The Speed values will be one of the following: S = “Slow Start” M = “Medium Start” F = “Fast Start” E.g. S/444 = Slow start and was placed 4th at each of the races three split points.	No
Split	This is a decimal value to two decimal places representing a dogs time at the first split marker	No
Checks	Whether the dog was checked in running by other dogs, and the number of lengths lost as a result of the checking, ie. C1	No
Vet	Whether a vetting period was imposed. Proceeded by “V”, followed by an number indicating the stand-down period. Ie. V21	No
Running	Whether it particularly ran wide “W”, or on the rails “R”.	No
RaceNumber	The number of the race which makes up the form. It has a ‘R’ preceding the number	No
Margin1	This is a decimal value to two decimal places representing a dogs margin from the winning dog, in the case of the winning dog, it is the margin to the second dog. Refer to the section “Margin Conversions”	No
Margin2	This is a decimal value to two decimal places representing a dogs margin from the dog in front if it, in the case of the winning dog this value is empty. Refer to the section “Margin Conversions”	No
SplitMargin	This is a decimal value to two decimal places representing a dogs time at the first split marker. Refer to the section “Margin Conversions”	No
RunTime	This is a decimal value to two decimal places representing a dogs running time for a race	No
Split1Time	This is a decimal value to two decimal places representing the time of the lead dog at the first split	No
Split1Rug	This is an integer between 1 and 10	No
Split2Time	This is a decimal value to two decimal places representing the time of the lead dog at the second split	No
Split2Rug	This is an integer between 1 and 10	No
Split3Time	This is a decimal value to two decimal places representing the time of the lead dog at the third split	No
Split3Rug	This is an integer between 1 and 10	No

Data Type Name	Description	Required
WinDiv	This is a decimal value of the winning dividend > 0 that is prefixed by the character "$".	No
Place1Div	This is a decimal value of the first place dividend > 0 that is prefixed by the character "$".	No
Place2Div	This is a decimal value of the second place dividend > 0 that is prefixed by the character "$".	No
Place3Div	This is a decimal value of the third place dividend > 0 that is prefixed by the character "$".	No
Quin	This is a decimal value of the quinella dividend > 0 that is prefixed by the character "$".	No
Exacta	This is a decimal value of the exacta dividend > 0 that is prefixed by the character "$".	No
Trifecta	This is a decimal value of the trifecta dividend > 0 that is prefixed by the character "$".	No
Pick4	This is a decimal value of the first four dividend > 0 that is prefixed by the character "$".	No
RD	This is a decimal value of the running double dividend	No
DD	This is a decimal value of the daily double dividend	No
Quad	This is a decimal value of the quaddie dividend	No
MeetingCode	This is a string value consisting of the track code and date a trail was held	Yes
FastOnNight	Yes/No	Yes
TrialReason	String value. Possible values are: Satisfactory Weight Whelping Exhibition	Yes

7.1. Margin Conversion

Where a margin value is greater than or equal to 1 dog length then the value will be displayed as a decimal value to 2 decimal places, suffixed with the character "L". Under 1 dog length the values must be displayed as the following abbreviation.

Length Type	Abbreviation	Equivalent Dog Length
Nose	NS	0.0L - 0.01L
½ Head	HHD	0.011L – 0.02L
Head	HD	0.021L – 0.05
½ Neck	HNK	0.051L – 0.1L
Neck	NK	0.11L – 0.25L
½ Length	HLH	0.26 – 0.5L
¾ Length	3LH	0.51L – 0.75L

8. End To End Scenarios

Scenario	Description	Outcome
A valid request is made to download data	Given a valid security key, valid access credentials and a valid track and date.	The request response status code is "200 OK" and the response contains either formatted XML or CSV data.
No security key specified	The URL request type requires a security key, and no key was specified.	The Http response status code is "412 Precondition Failed". The response body would contain "<?xml version="1.0"?> <exception>No Security Key Specified</exception>"
Invalid security key specified	The URL request type requires a security key, and the security key does not return a commercial user account reference	The Http response status code is "412 Precondition Failed". The response body would contain "<?xml version="1.0"?> <exception>Invalid Security Key</exception>"
Inadequate access credentials	The URL request type requires a security key, and the security key supplied returns a commercial user with inadequate access rights.	The Http response status code is "412 Precondition Failed". The response body would contain "<?xml version="1.0"?> <exception>Access Denied</exception>"
Invalid track code	The URL request type requires an NDR Track code, and the track code supplied does not reference a valid track in our system.	The Http response status code is "412 Precondition Failed". The response body would contain "<?xml version="1.0"?> <exception>Invalid Track Code</exception>"
Invalid Date	The URL request type requires a date to identify a meeting, and the date supplied is not in a valid format.	The Http response status code is "412 Precondition Failed". The response body would contain "<?xml version="1.0"?> <exception>Invalid Date Specified</exception>"
Invalid meeting	The URL request type requires a date and NDR track code to identify a meeting but a valid meeting cannot be found on that date at the track specified.	The Http response status code is "412 Precondition Failed". The response body would contain "<?xml version="1.0"?> <exception>Not a Valid Meeting</exception>"

Scenario	Description	Outcome
Invalid File type	The URL request type requires a file type to be supplied and the format cannot be identified.	The Http response status code is "412 Precondition Failed". The response body would contain "<?xml version="1.0"?> <exception>Invalid File Type Specified</exception>"

9. Track Codes

The following table gives the list of track codes within FastTrack.

Track Code	State	Full Track Name	State Code
		NEW SOUTH WALES	
200	**NSW**	**Wentworth Park**	**WPK**
201	NSW	Dapto	DTO
202	**NSW**	**Bulli**	**BUL**
203	NSW	Casino	CAS
204	**NSW**	**Cessnock**	**CES**
205	NSW	Gosford	GOS
206	**NSW**	**Grafton**	**GRA**
207	NSW	Lismore	LIS
208	**NSW**	**Maitland**	**MAT**
209	NSW	Muswellbrook	MUS
210	**NSW**	**Newcastle**	**NCL**
211	NSW	Nowra	NOW
212	**NSW**	**Orange**	**ORG**
213	NSW	Penrith	PEN
214	**NSW**	**Richmond**	**RIC**
215	NSW	Moree	MOR
216	**NSW**	**Wyong**	**WYG**
217	NSW	Singleton	SIG

Track Code	State	Full Track Name	State Code
218	**NSW**	**Taree**	**TAR**
219	NSW	Temora	TEM
220	**NSW**	**Wagga Wagga**	**WAG**
221	NSW	Wauchope	WAU
222	**NSW**	**Tweed Heads**	**TWH**
223	NSW	Albury	ALB
224	**NSW**	**Appin**	**APP**
225	NSW	Armidale	ARM
226	**NSW**	**Bathurst**	**BAT**
227	NSW	Broken Hill	BHL
228	**NSW**	**Coonabarabran**	**CON**
229	NSW	Coonamble	COO
230	**NSW**	**Cootamundra**	**COT**
231	NSW	Dubbo	DUB
232	**NSW**	**Forbes**	**FOR**
233	NSW	Goulburn	GBN
234	**NSW**	**Griffith**	**GRF**
235	NSW	Gunnedah	GUN
236	**NSW**	**Harold Park**	**HPK**
237	NSW	Kempsey	KEM
238	**NSW**	**Lithgow**	**LIT**

Track Code	State	Full Track Name	State Code
239	NSW	Moss Vale	MVL
240	**NSW**	**Mudgee**	**MUD**
241	NSW	Queanbeyan	QBN
242	**NSW**	**Tamworth**	**TAM**
243	NSW	Wollongong	WOL
244	**NSW**	**Young**	**YNG**
245	NSW	Cowra	COW
246	**NSW**	**Wentworth**	**WEN**
247	NSW	Narrabri	NAR
248	**NSW**	**Potts Park**	**PPK**
249	NSW	The Gardens	GAR
250	**NSW**	**Canberra**	**ACT**
		VICTORIA **** Provincial meeting held at metropolitan track**	
300	**VIC**	**The Meadows**	**MEA**
315	**VIC**	**The Meadows**	**MEP ****
301	VIC	Sandown Park	SAN
314	VIC	Sandown Park	SAP **
302	**VIC**	**Ballarat**	**BAL**
303	VIC	Bendigo	BEN
304	**VIC**	**Cranbourne**	**CRN**
305	VIC	Geelong	GEL

Track Code	State	Full Track Name	State Code
306	**VIC**	**Healesville**	**HVL**
307	VIC	Horsham	HOR
308	**VIC**	**Shepparton**	**SHP**
309	VIC	Sale	SLE
310	**VIC**	**Traralgon**	**TRA**
311	VIC	Wangaratta	WTA
312	**VIC**	**Warragul**	**WGL**
313	VIC	Warrnambool	WBL
QUEENSLAND			
400	QLD	Brisbane	BGC
401	**QLD**	**Beenleigh**	**BEN**
402	QLD	Bundaberg	BUN
403	**QLD**	**Cairns**	**CAI**
404	QLD	Capalaba	CAP
405	**QLD**	**Gabba**	**GAB**
406	QLD	Gold Coast	GCT
407	**QLD**	**Ipswich**	**IPS***
408	QLD	Lawnton	LAW
409	**QLD**	**Ayr**	**AYR**
410	QLD	Mackay	MCK
411	**QLD**	**Mt Isa**	**MTI**

Track Code	State	Full Track Name	State Code
412	QLD	Rockhampton	ROC
413	**QLD**	**Townsville**	**TOW**
414	QLD	Toowoomba	TWB
		SOUTH AUSTRALIA	
500	SA	Angle Park	APK , APD , APM
501	**SA**	**Barmera**	**BAR**
502	SA	Gawler	GAW
503	**SA**	**Kulpara**	**KUL**
504	SA	Mount Gambier	MTG
505	**SA**	**Port Augusta**	**PTA**
506	SA	Port Lincoln	PTL
507	**SA**	**Port Pirie**	**PTP**
508	SA	Strathalbyn	STR
509	**SA**	**Whyalla**	**WHY**
		WESTERN AUSTRALIA	
600	**WA**	**Cannington**	**CAN**
601	WA	Mandurah	MAN
602	**WA**	**Northam**	**NOR**
		TASMANIA	
700	**TAS**	**Hobart**	**HOB**
701	TAS	Launceston	LCN

Track Code	State	Full Track Name	State Code
702	**TAS**	**Devonport**	**DEV**
NORTHERN TERRITORY			
800	**NT**	**Darwin**	**DAR**
NEW ZEALAND			
900	**NZ**	**Christchurch**	**CCH**
901	NZ	Wanganui	WAN
902	**NZ**	**Manawatu**	**MAW**
903	NZ	Auckland	AUK
904	**NZ**	**Waikato**	**WAK**
905	NZ	Otago	OTG
906	**NZ**	**Southland**	**SOU**
907	NZ	Wairarapa	WAI
908	**NZ**	**Wellington**	**WEL**
909	NZ	Tokoroa	TOK
910	**NZ**	**Taranaki**	**TAK**
911	NZ	Palmerston North	PNN
912	**NZ**	**Ashburton**	**ASH**