New JCapper Program Update 05/30/2012

What's Inside –

WagerHistory Module –

WagerHistory Table – VetScratchDays Data Field
The JCapper2.mdb file in this program update contains a data field change - meaning that it is in your best interest to check the box on the face of the Installer to make it give you a new JCapper2.mdb file (and then import data out of your old file as the last step performed as part of the install.) The TimeStamp field, which was not being used, has been replaced with a new data field named VetScratchDays. This new data field is designed to hold a numeric value (data type long integer) indicating the number of days since a horse's most recent vet scratch. The default value for this field is 0 (which is used to indicate a horse that has not recently been listed as a vet scratch.)

WagerHistory Module Data Entry Sheet –

Startup Behavior – Auto Generation of "Dummy Wager"
Beginning with the release of this program update, upon startup, the WagerHistory Module Data Entry Sheet tests the WagerHistory table and counts the number of table records. If at least 1 table record for a wager is found in the table, the module loads (and runs) normally. If 0 table records are found, instead of the BOF message, and instead of the user having to manually enter a "dummy wager" as the 1st table record - here's what happens:

1. The user is notified that 0 table records exist and is then prompted: "Do you wish to Create a "dummy wager" as the 1st table record Y/N?"

2. Upon answering Y at the prompt, the interface auto generates table record number 1 with a made up track code, race number, and horse name - and the module loads normally from there.

Hint: After using the WagerHistory Module to record at least one real wager, consider deleting the "dummy" wager.

New Data Items

VetScratchDays
Beginning with this program update, when the Data Entry Sheet interface is used to auto populate the UDM Count data fields, two new data items are created. The first is a factor unique to the WagerHistory Module called VetScratchDays. This is the number of days going back in time from today's race to the most recent VetScratch entry for the horse.

The default value for this factor is 0 (which is used to indicate a horse that has not recently been vet scratched.) Beginning with this program update, when a new record is saved, the number of VetScratchDays is written to the VetScratchDays field of the WagerHistory table. This has the advantage of enabling query results in the WagerHistory Module's Reporting Engine to be broken out by VetScratchDays.

XML UDM Profile
The second data item is an XML text string which contains the horse's entire UDM profile. The opening tag name is <UDMProfile>. The closing tag name is <\UDMProfile>. Individual UDM names are wrapped with XML tags having the names <UDM> (opening tag) and <\UDM> (closing tag.)

Beginning with this program update, when you save a new WagerHistory table entry for a single horse, the entire UDM profile for that horse is written to the Notes field in the WagerHistory table.

This has the advantage of enabling you to write and execute stored sql expressions to filter WagerHistory query results by individual UDM names. (Specific examples of sql expressions for doing this are provided below in the section about the Reporting Engine.)

Reporting Engine – New Data Items

VetScratchDays
VetScratchDays has been added to the factors drop down – enabling you to break your query results out by the number of vet scratch days.

XML UDM Profile
Having the entire UDM Profile available for individual horses stored as text XML in the Notes field of the WagerHistory table means that you can write sql expressions to generate reports filtered by UDM names – as well as combinations of UDM names.

Add the following line to the sql expression driving your report if you want to restrict the query results to horses flagged by an A UDM named 0_A-MyUDM:

AND INSTR(NOTES, '0_A-MyUDM') > 0

Note that above line causes an evaluation between the Notes field and the UDM name wrapped in single quote characters. Also note that the evaluation is NOT (repeat not) case sensitive.

To restrict a report to horses not flagged by the same (specific) UDM, change the greater than zero part of the expression to equal to zero – like this:

AND INSTR(NOTES, '0_A-MyUDM') = 0

Note that multiple lines can be used in the sql expression driving your reports. To make the report contain query results for horses selected by an A UDM named 0_A-MyUDM but not horses selected by a negative expectation UDM named x_NoTalentTrainers, use the following two lines:

AND INSTR(NOTES, '0_A-MyUDM') > 0
AND INSTR(NOTES, 'x_ NoTalentTrainers') = 0

Hint: With a little thought and creativity, you can write sql expressions that will enable you to discover the interplay amongst the UDMs flagging the horses in your wager history.

Utilities Screen –

XML UDM Profile Button
The Utilities Interface for the WagerHistory Module shipped with this program update contains a new button. The new button is simply labeled XML UDM Profile. When this button is clicked, the following chain of events is executed:

1. The interface will process each record in the WagerHistory table.

2. If a UDM Profile has been stored for the current horse being processed, the complete XML UDM Profile for that horse is created and then written to the Notes field.

3. If the horse for the current record being processed is flagged by the auto generated x_VetScratch UDM, the number of days since the most recent vet scratch is calculated and written to the VetScratchDays field.

4. The next record is pulled up and the process is repeated until every record in the WagerHistory Table has been processed.

You need to have a current WagerHistory table (one that has a VetScratchDays data field as described above) to run this routine. New records created with the WagerHistory Module shipped as part of this program update will have values populated for VetScratchDays and XML UDM Profile in the WagerHistory Table. However, older pre-existing records in your WagerHistory table do not (yet) have values for these data items.

Clicking the XML UDM Profile button and running this routine will populate the Notes and VetScratches fields in your WagerHistory table with data – as much as it is possible to do so based on what you stored in the table at the time each initial table entry was created.

HTML Report Race Summary Saddle Cloth Numbers
In this JCapper Program Update, I have rewritten the algorithm that generates the saddle cloth numbers displayed as part of the race summary. The key difference is that the new algorithm recognizes ties (whereas the old algorithm did not.) Here is a simple example:

Horse/Number/Rank
#1 85 (4)
#2 78 (5)
#3 91 (1)
#4 91 (1)
#5 87 (2)

The old algorithm would display the saddle cloth numbers as follows: 3-4-5-1

The new algorithm uses the comma character to indicate a tie and would display the saddle cloth numbers as follows: 3,4-5-1

Hint: The presence of a comma character means that a tie in rank exists.

Enhanced Settings Module - New Settings

Expanded STAT Handling
Ron Tiller at HDW recently added some new data items to the HDW V4 JCapper binary race file. The new data items (so far) are:
E2 Race Par – each of the last 10 running lines
F3 Race Par – each of the last 10 running lines
HDW Speed Fig Par – each of the last 10 running lines

While I am not (yet) doing anything with these new data items in terms of the JCapper factor set, I have created a new setting (an ON-OFF switch actually) in the Enhanced Settings area of the program. The new setting is labeled as Expanded STAT Handling and it behaves in the following manner:

0-OFF - When this (default) option is persisted in the Enhanced Settings Module, the expanded stat data items ARE NOT inserted into .JCP files during a .JCP file build routine.

1-ON - When this option is persisted in the Enhanced Settings Module, the expanded stat data items ARE inserted into .JCP files during a .JCP file build routine.

I know that many of you are importing .JCP files into your own third party apps and spreadsheets. My purpose for creating an ON-OFF switch in the Enhanced Settings area is to not “break” any third party apps you may be using to handle .JCP files in their

current file format. The ON-OFF switch gives you control over the format of your .JCP files. Use the OFF switch in the event you don't want the new data items. Conversely, the ON switch provides you a mechanism to start inserting the new data items into .JCP files now (ahead of any future development I might do with these in terms of getting them added to the JCapper factor set.)

One other note about the expanded STAT area: In the coming weeks Ron should be adding more data items for us. The same ON-OFF switch discussed here will apply to those as well. (Stay tuned.)

Rider Trainer A/E Rating History Type

I have recently been experimenting at my end with an A/E (Actual vs. Expected) CXN Rating. Long story short, it relates to riders and trainers whose starters outperform – or achieve a greater number of actual winners than what you'd expect from the odds of their starters. The formula for arriving at the resulting rating is proprietary. (And it may be subject to future changes based on R&D going forward.) But I am comfortable enough with results seen so far that I have created an enhanced setting that gives the user complete control over WHERE that formula gets applied.

That said, here are the options for the new setting:

0 None – This (default) setting causes A/E Ratings for both rider and trainer NOT to be generated at .JCP file build time. (Use this setting for max speed when generating .JCP files – and to prevent the A/E ratings for rider and trainer from being generated and written to JCP files in the first place.) This setting also causes the PPs Generator to NOT display A/E Ratings for rider and trainer.

7 Day History – This setting causes A/E Ratings for both rider and trainer to be generated at .JCP file build time for the most recent 7 calendar days. The routine that generates A/E ratings for rider and trainer will take the time to query your c:\JCapper\Exe\JCapper2.mdb file StarterHistory table for each rider and trainer name found in the .JCP file. That means that your StarterHistory table must contain the needed history data. Note that the resulting A/E Rating for Rider and/or Trainer will be generated using whatever history data is sitting in the table at .JCP file build time. (If for example, the table is empty, the resulting A/E Ratings will be zero for both rider and trainer.)

15 Day History – This setting behaves exactly the same as the 7 day history setting – EXCEPT that the 15 most recent calendar days are used when A/E Ratings for rider and trainer are generated at JCP file build time.

21 through 365 Day History Settings - These settings behave exactly the same as the 7 day history setting – EXCEPT that the number of calendar days from the persisted setting itself are used when A/E Ratings for rider and trainer are generated at JCP file build time.

-365 One Year History Current Track – This setting causes A/E Ratings for both rider and trainer to be generated at .JCP file build time for the most recent 365 calendar days at the same track code as the .JCP file being built. The routine that generates A/E ratings for rider and trainer will take the time to query your c:\JCapper\Exe\JCapper2.mdb file StarterHistory table for each rider and trainer name found in the .JCP file. That means that your StarterHistory table must contain the needed history data. Note that the resulting A/E Rating for Rider and/or Trainer will be generated using whatever history data is sitting in the table at .JCP file build time. (If for example, the table is empty, the resulting A/E Ratings will be zero for both rider and trainer. Or, if the StarterHistory table contains fewer than the required number of days, the A/E Rating generated will be based on whatever history exists in the table at .JCP file build time.)

Warning about .JCP File Build Routine Speed
Using a setting other than 0 None causes the .JCP file build routine to perform an additional step. As the name for each rider and trainer is encountered as part of the file build process, the interface will create a sql expression based on your chosen setting and then execute that sql expression to query the StarterHistory table in your c:\JCapper\Exe\JCapper2.mdb file. This step is necessary in order to tabulate data required for the A/E Rating calculation.

However, this step takes extra time and causes the .JCP file build routine to slow noticeably.

In a future program update, I will be adding a Halt .JCP File Build Routine button to the HDW File Manager User Interface. The purpose for having such a button would be to allow the user who accidentally initiated a .JCP file build routine spanning a significant time period (a month, a quarter, or even an entire year) – but then realized too late that the routine would take too long to complete. By clicking the button, the routine could be halted – and the user could come back and rerun it later at a more convenient time.

Why A/E Ratings as opposed to win rate and roi?
Everybody can look at rider and trainer stats and see number of starters, win rate, and flat win bet roi. I wanted to create something that gives the JCapper player a slightly different look. Some riders and trainers are CRAFTY. CXN stats can be skewed for both rider and trainer because of the sheer number of starters who are no hopers (horses that really should be 150-1 or more in the odds to win the race.) But every now and then (when they actually have one that is well meant) that well meant starter outruns its odds for them. The A/E Rating is an attempt at spotting connections with a proven penchant for pulling off upsets.

PPs Generator

The Rider Trainer A/E Rating History Type settings discussed above for the Enhanced Settings Module also control what you now see just to the right of names for riders and trainers in JCapper past performances.

0 None – This setting causes A/E Ratings for rider and trainer NOT to be displayed in JCapper past performances.

All other settings cause A/E Ratings for rider and trainer based on the persisted setting itself to be displayed in JCapper past performances – PROVIDED that the .JCP file being used to render past performances contains the necessary data in fields 26 and 27.

Note that the A/E Ratings displayed to the right of rider and trainer names in JCapper Past Performances are determined by data file content read from field numbers 26 and 27 in the .JCP file. Changing the Rider Trainer A/E Rating History Type setting in the Enhanced Settings Module prior to generating past performances will in no way shape or form affect content sitting in .JCP files that have been previously built. To make a change to the A/E Ratings Type displayed in past performances, you must make the appropriate settings change in the Enhanced Settings Module, rebuild .JCP files using the new setting, and then generate past performances using a rebuilt .JCP file.

Scratch BOT

Hollywood Park
The Scratch BOT module found in this program update contains new logic that recognizes BHP as a valid tote code for Hollywood Park (or Betfair Hollywood Park) for those of you who are using Brisnet files. (Note: HDW data subscribers were unaffected by the change in track code.)

Equibase recently began listing Hollywood Park as BETFAIR HOLLYWOOD PARK in the XML. This change does not necessitate an internal programming change in Scratch BOT. The correct Notes Field entry for parsing scratches and changes from the XML for Hollywood Park simply becomes: BETFAIR HOLLYWOOD PARK

Charles Town
Recently, Equibase began listing Charles Town as HOLLYWOOD CASINO AT CHARLES TOWN RACES in the XML. Earlier versions of Scratch BOT were based on the info model that Charles Town would be a valid Notes Field table entry in the TrackAbbreviations table no matter how the track_name for Charles Town might be listed in the XML. To keep the same info model intact, I made an internal programming change in Scratch BOT for parsing the new way Equibase has the track_name listed in the XML. The Scratch BOT Module in this program update contains new XML parsing logic that allows you to continue to using CHARLES TOWN as a valid track name in the Notes Field of the TrackAbbreviations table if that is your choice. Alternately,

HOLLYWOOD CASINO AT CHARLES TOWN RACES is also a valid table entry. (Both are valid table entries. Pick one and go with it.)

Live Play Module – BHP Tote Code
The Live Play Module found in this program update contains new logic that recognizes BHP as a valid tote code for Hollywood Park (or Betfair Hollywood Park) for those of you who are using Brisnet files. (Note: HDW data subscribers were unaffected by the change in track code.)

DFM Card Loader

Expandable User Interface
Beginning with this program update, the user interface for The DFM Card Loader is now expandable. If you grab the lower edge of the module border with your mouse cursor and then pull the border down to expand the module – the file and folder list boxes will expand with the module itself. Summer is coming, and this change in season always brings an increase in the number of tracks that are running. The ability to expand the file and list boxes on the interface should make it easier to see the card files you are working with.

Post Time Reporting - R1 for Selected Card File
The DFM Card Loader has a new button located above the available files list box on the left hand side of the module. The button is labeled as 1ST Post. When this button is clicked, logic in the module will parse the first selected card file in the available files list box. After parsing the file, a message box appears. The message box contains a simple report that displays post time for the first race found in the selected file. The advent of summer brings an increase in the number of tracks running each day. Many of them have post times for R1 that vary from one day to the next. It can be tough (if not impossible) to stay on top of when R1 is slated to go off at each track – especially if you are playing multiple tracks. Adding a button to enable a quick file parse that generates a pop up report displaying post time for the first race in a selected file should make it easier for those of you playing multiple tracks to (quickly) determine where on the interface (which folder) any given file should be dragged and dropped to.

FAQs:

Q. Do I need to check the box on the Installer to make it give me a new JCapper2.mdb

file?

A. Yes. This new program update contains new JCapper2.mdb table content - which requires that you make the Installer give you a new JCapper2.mdb file by checking the Overwrite JCapper2.mdg file box on the face of the Installer.

Q. After checking the box to make the Installer give me a new JCapper2.mdb file, do I need to use the JCapper2 Import Module found on the User System Definitions Screen to import data from my old JCapper2.mdb file into the fresh blank JCapper2.mdb file copied onto my c:\JCapper\Exe folder by the Installer?

A. Yes, any time you make the installer give you a new JCapper2.mdb file, this is critical - an absolutely must - especially if you have a custom sql factor setup or a custom sql html report.

Q. How do I run a JCapper2 Import routine?

A. Link to video:

http://www.jcapper.com/podcasts/j2import.wmv

Note: I generally recommend checking the Overwrite JCapper2.mdb File box on the face of the Installer as a best practice each time you install a new program update.

I also recommend importing data out of you old JCapper2.mdb file into your new file too.

If you get in the habit of doing this with each new program update, you're never left guessing whether or not the new program update you just installed contains new required table content. Checking the box makes the Installer give you new table content whenever it's there.

Q. Do I need to rebuild databases from scratch after installing this program update?

A. No. Not if you've kept current with the downloads.

Now if you haven't kept current with the downloads then yes, chances are you will have to rebuild databases from scratch before using the Data Window to query playlist files built using older program versions.

Enjoy,

-jp

.