DateTime Slot Values in RiverWare

Use Demo, Development Requirements and Internal Design

Author: Phil Weinstein

This document describes support in RiverWare 4.9 for DateTime values in Series, Table and Scalar Slots.

Fully specified Date/Times (down to minute resolution), plus several "Partial" DateTime formats are supported. DateTime values, including data indicating which of the DateTime "Parts" are relevant, are encoded into the Slot's double-precision floating point values.

A new RiverWare Unit Type is used to distinguish DateTime values from conventional numeric values. Eight different Units within the new DateTime Unit Type are used to distinguish the different supported "Partial" DateTime formats.

Predefined RPL Functions have been added to convert "double-encoded" Partial DateTime slot values to and from RPL Symbol DateTime values.

The analysis and design outlined in this document supersedes a prior analysis for "Non-numeric SeriesSlots in RiverWare" (1-19-2006, Phil Weinstein, 14 pages).

0.1 Document Status

4-17-2007: Post-development revisions.
4-18-2007: minor edits and additions.

0.2 Contents

1.0 User and RPL Interface Overview

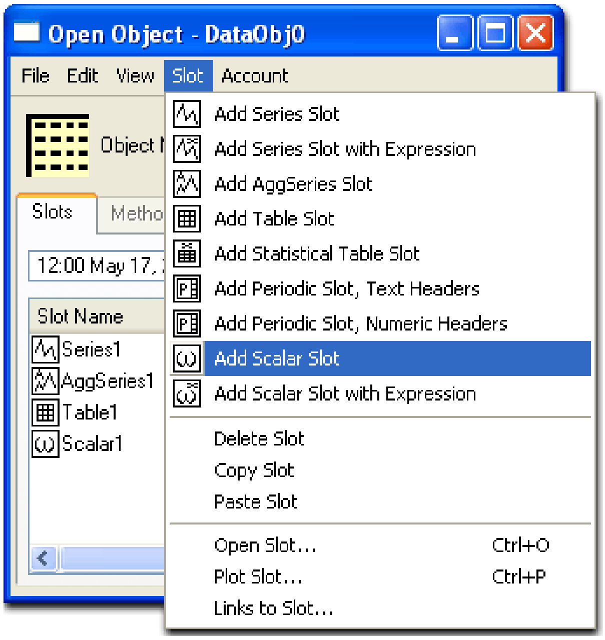
1.1 Walkthrough: Adding a Slot with DateTime Values to a Data Object

To introduce how DateTime Values exist on RiverWare Slots, this section walks the user through the process of creating a Series Slot with DateTime values on a Data Object.

STEP 1

Create a Data Object and **Add a Series Slot,** or any of the following types of Slots. All of these support DateTime values.

- Series Slot
- AggSeries Slot
- Table Slot
- Scalar Slot

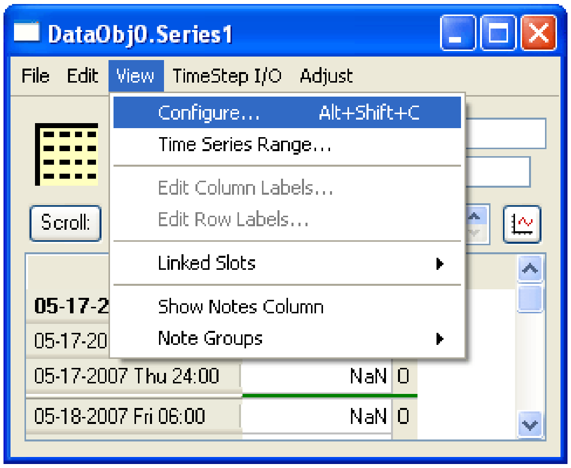


STEP 2

From the Open Object Dialog, **double click on your new Series Slot** to show the Open Series Slot dialog.

STEP 3

From the Open Series Slot's **View menu,** select **"Configure..."** to open the Series Slot Configuration dialog box ....

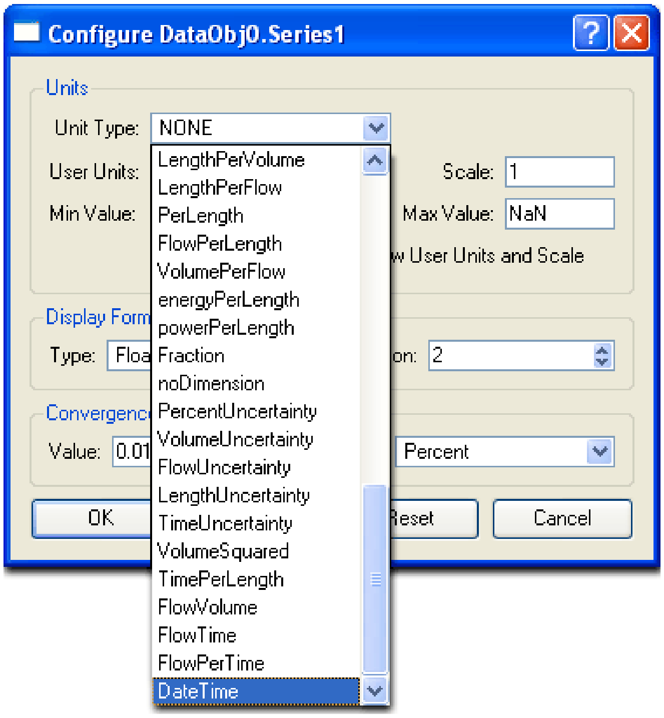


STEP 4

From the Series Slot Configuration dialog box, select the **last item** from the **Unit Type** option menu: **"DateTime".**

The "DateTime" Unit Type, and the eight supported DateTime Units are special in RiverWare. These selections cause the Slot's values to be displayed and edited as DateTimes rather than as numeric values.

Unlike switching to an ordinary Unit Type, switching to the "Date Time" unit type, and between the various DateTime "units" causes a change in the underlying Slot values. The user must confirm these changes with this popup dialog.



Configure DataObj0.Series1

Units

Unit Type: NONE

User Units:

Min Value:

Scale: 1

Max Value: NaN

Display Form

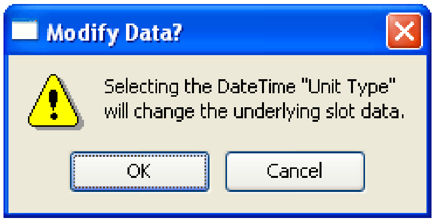
Type: Floa

on: 2

Convergenc

Value: 0.01

Percent



Modify Data?

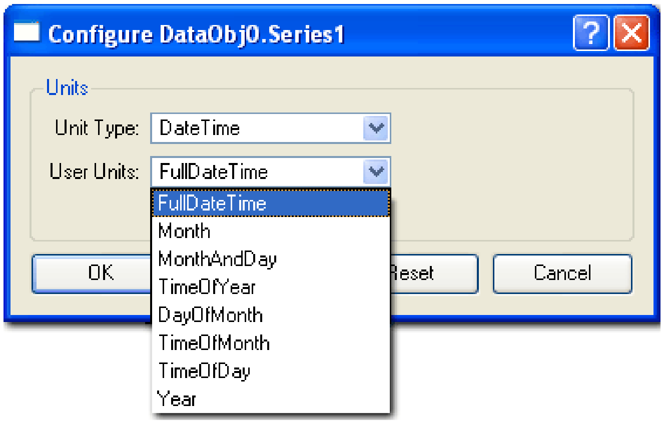
Selecting the DateTime "Unit Type" will change the underlying slot data.

STEP 5

As a result of switching to the "DateTime" Unit Type, configuration properties associated with only numeric values are hidden (e.g. Min/Max values, Display Format, Convergence).

Notice the eight different "User Units" associated with the DateTime Unit Type. You can keep the default selection, "FullDateTime" ... **Click OK.**

Note: AggSeries Slots and Table Slots support independent Numeric / DateTime configuration on each column.



Configure DataObj0.Series1

Units

Unit Type: DateTime

User Units: FullDateTime

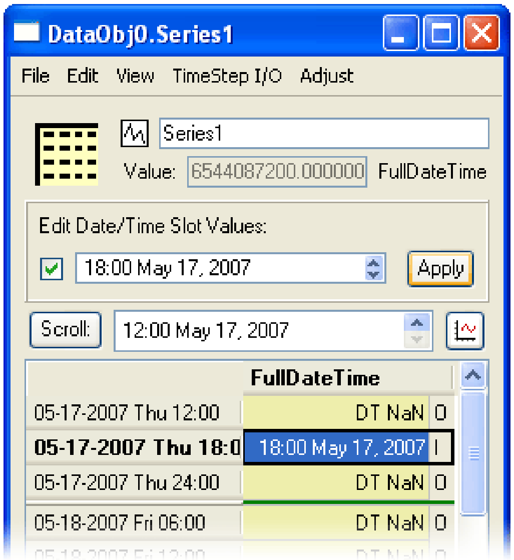
STEP 6

As a result of selecting a "DateTime" unit, a **Partial DateTime Editor** is added to the Open Slot dialog. (See the box containing the "Apply" button).

The entry and selection fields shown within the Partial DateTime Editor depend on which DateTime "User Unit" selection was made. For the **FullDateTime** selection, two alternative entry modes are available, selectable with the checkbox on the left.

- **Timestep Spinner** -- for DateTimes on Timestep Boundaries. (first image).
- **Unconstrained Date/Times** -- for selection of any DateTime within the supported RiverWare date range. (second image) ....

Try making different **Cell Selections,** hitting the **"Apply" button** and repeating these steps with different DateTime entry values.



Series1

Value: 6544087200.000000 FullDateTime

Edit Date/Time Slot Values:

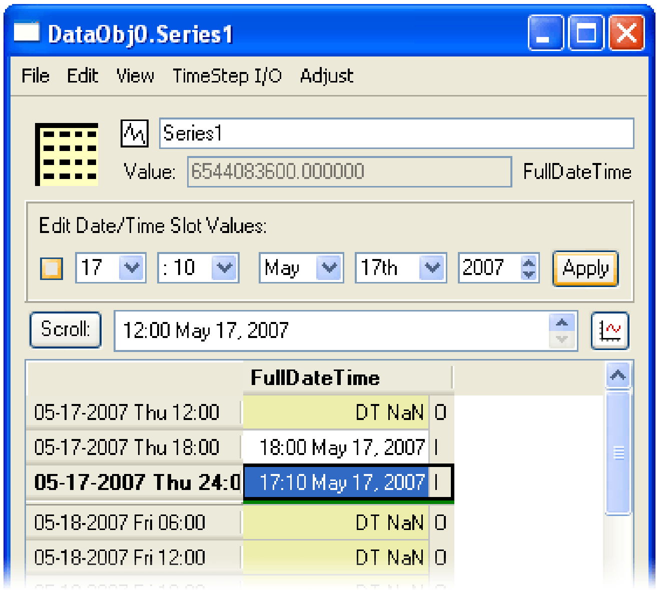
☑ 18:00 May 17, 2007

Scroll: 12:00 May 17, 2007

	FullDateTime	
05-17-2007 Thu 12:00	DT NaN	0
05-17-2007 Thu 18:0	18:00 May 17, 2007	I
05-17-2007 Thu 24:00	DT NaN	0
05-18-2007 Fri 06:00	DT NaN	0

Notice that the "Value" field at the top of the Open Slot dialog is disabled (not editable, and shown with a gray background). This field shows the actual internal numeric value associated with the selected cell's DateTime value. (For Slots configured for conventional numeric values, the Value field is editable).

Double-clicking on a cell assigns the cell's value to the Partial DateTime Editor.



Series1

Value: 6544083600.000000 FullDateTime

Edit Date/Time Slot Values:

☐ 17 : 10 May 17th 2007

Scroll: 12:00 May 17, 2007

	FullDateTime	
05-17-2007 Thu 12:00	DT NaN	0
05-17-2007 Thu 18:00	18:00 May 17, 2007	I
05-17-2007 Thu 24:0	17:10 May 17, 2007	I
05-18-2007 Fri 06:00	DT NaN	0
05-18-2007 Fri 12:00	DT NaN	0

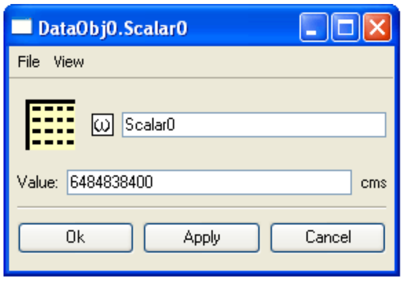
When a range of cells is selected with the first and last cell having valid values, interpolated DateTimes can be assigned to the intervening cells. Try: **Edit** >> **Interpolate** (not illustrated here).

Plotting of DateTime value series is minimally supported. (Hit the **Plot icon button** on the right side of the dialog). DateTime coordinates are not shown on the axes, but relative DateTime "magnitudes" can be discerned.

DateTime values which don't conform to the Slot's (or Slot column's) configured DateTime "units" (Partial DateTime configuration) are shown with a **yellow cell background.**

1.2 Partial DateTime Editor for Scalar Slots

The Partial DateTime Editor in the Open ScalarSlot dialog also shows the applied DateTime. This page shows this dialog in standard numeric configuration and in the various Full and Partial DateTime configurations.

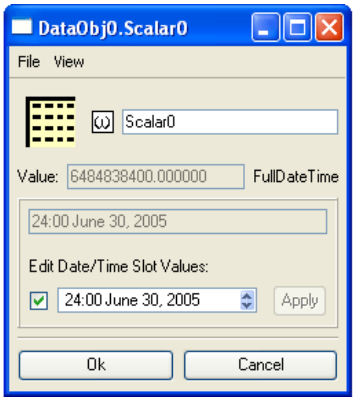


Scalar0

Value: 6484838400 cms

Numeric (standard) Scalar Slot

FullDateTime Scalar Slot >>



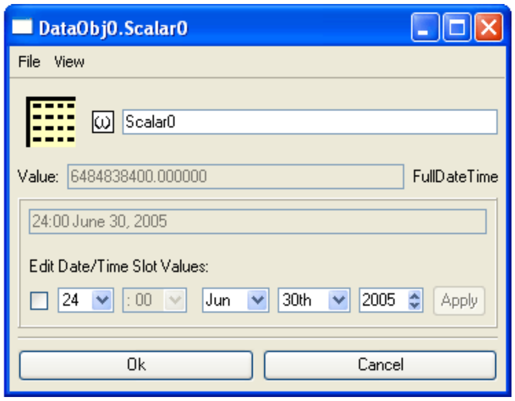
Scalar0

Value: 6484838400.000000 FullDateTime

24:00 June 30, 2005

Edit Date/Time Slot Values:

☑ 24:00 June 30, 2005



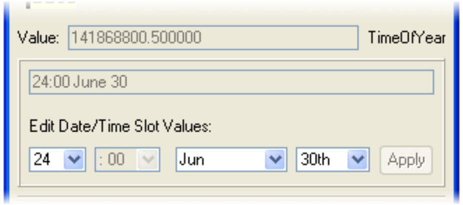
Scalar0

Value: 6484838400.000000 FullDateTime

24:00 June 30, 2005

Edit Date/Time Slot Values:

□ 24 : 00 Jun 30th 2005



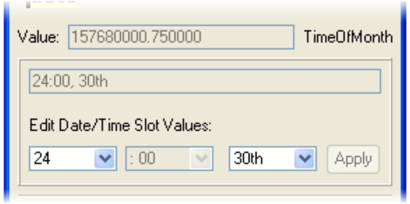
Value: 141868800.500000 TimeOfYear

24:00 June 30

Edit Date/Time Slot Values:

24 : 00 Jun 30th

Editors for Partial DateTimes on Scalar Slots

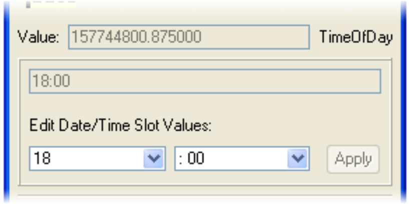


Value: 157680000.750000 TimeOfMonth

24:00, 30th

Edit Date/Time Slot Values:

24 : 00 30th

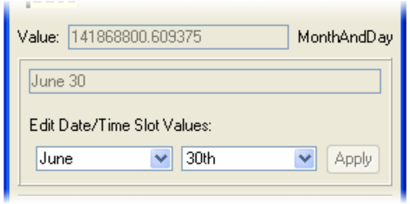


Value: 157744800.875000 TimeOfDay

18:00

Edit Date/Time Slot Values:

18 : 00

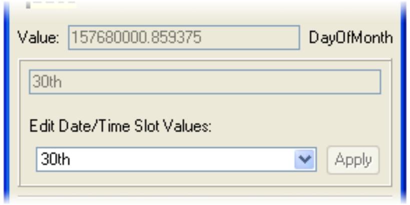


Value: 141868800.609375 MonthAndDay

June 30

Edit Date/Time Slot Values:

June 30th

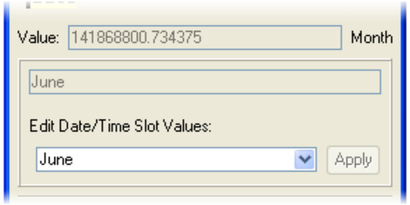


Value: 157680000.859375 DayOfMonth

30th

Edit Date/Time Slot Values:

30th

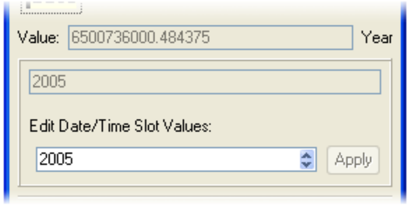


Value: 141868800.734375 Month

June

Edit Date/Time Slot Values:

June



Value: 6500736000.484375 Year

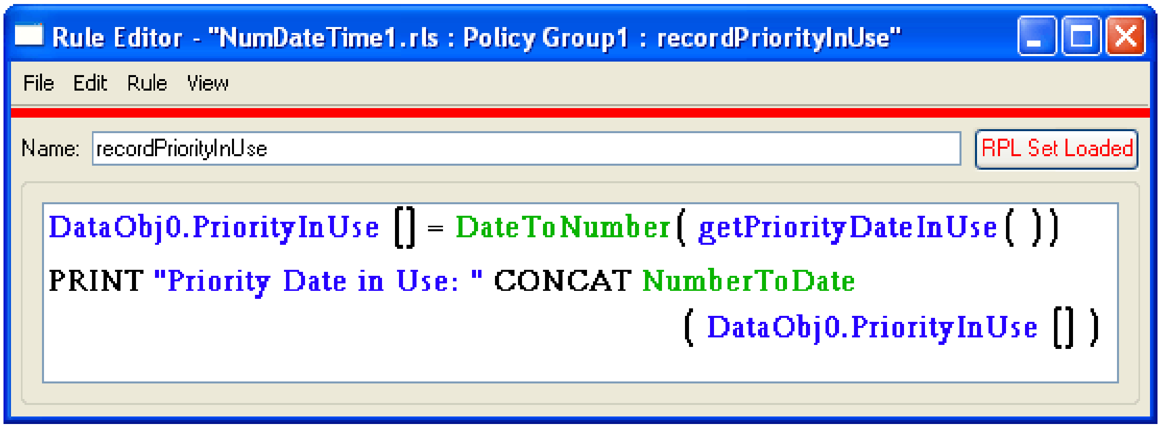
2005

Edit Date/Time Slot Values:

2005

1.3 Access to Slot DateTime values via RiverWare RPL

RPL doesn't directly support DateTime Slot values as DateTimes. All Slot values are handled as numeric values. But two new Predefined RPL Functions are used to convert the new Slot DateTime values (double-encoded partial date/times) to and from RPL DateTimes values. The following illustration demonstrates the use of the **DateToNumber** and **NumberToDate** Predefined functions:


Rule Editor - "NumDateTime1.rls : Policy Group1 : recordPriorityInUse"
File
Edit
Rule
View
RPL Set Loaded

Name: recordPriorityInUse

```
DataObj0.PriorityInUse [] = DateToNumber( getPriorityDateInUse( ))
PRINT "Priority Date in Use: " CONCAT NumberToDate
                                      ( DataObj0.PriorityInUse [] )
```

The following two sections provide more information about these predefined functions.

1.4 Predefined Function: DateToNumber

Description	Given a Date/Time value, returns that date encoded as a numeric value of the type used by slots to containing date/time values.	
Type	NUMERIC	
Arguments		
1	DATETIME	The date/time value to encode as a numeric value.
Comments	Slots representing date/time values have unit type DateTime. The date/time value need not be fully specified, but the return value should only be assigned to a slot with appropriate units. For example, if the value @”January 1” should only be assigned to a slot with units “MonthAndDay”.	

Syntax Examples:

```
DateToNumber(@”Current Timestep”)
```

1.5 Predefined Function: NumberToDate

Description	Given a numeric encoding of a date/time, returns the corresponding date/time value..	
Type	DATETIME	
Arguments		
1	NUMERIC	The numeric encoding of a date/time value.
Comments	Slots representing date/time values have unit type DateTime. Internally these values are represented as numbers although the interface displays them as date/times. Looking up a value on such a slot will retrieve the numeric encoding, this function converts that number to a date/time value as required to treat it as a date within policy. If the unit for the slot corresponds to a partially specified date/time format, then the result will a partially specified date/time value.	

Syntax Examples:

```
NumberToDate(Data.PriorityDate[])
```

might return @”January 12” if the Data.PriorityDate slot has units “MonthAndDay”.

2.0 Development Requirements

Prior to this enhancement, values within Series Slots, Table Slots and Scalar Slots were always represented and displayed as numeric values.

With this new work, numeric values in various types of Slots (internally represented with “double precision floating point values”) can represent one of several forms of date and time. This alternative representation of slot values is supported by special provisions within the GUI and by new RPL functions to map these values to and from RPL SymbolicDateTimes.

Support for “double-encoded” DateTime values will include the following provisions:

1. Support for Absolute Date/Times AND certain “Partial” Date/Times. Only those Partial Date/Times which are readily supportable with the RPL Symbolic Date/Time class will be supported as Slot values. This will include the following types of partial date/times:
   - Month
   - Month / Day of Month
   - Month / Day of Month / Hour / Minute / Second
   - Day of Month
   - Day of Month / Hour / Minute / Second
   - Hour / Minute / Second
   - Year

2. Support for Date/Times within the **full supported time range** within RiverWare (currently, between years 1800 and 3800).
3. Ability for the user to **edit Date/Time Slot values** via the GUI.
4. Ability encode and decode Date/Times as **RPL Symbolic Date/Times**.
5. Ability to save and load **model files** with Series Slots and Scalar Slots having Date/Time values.
6. Support for creation and configuration of Series Slots, Table Slots, and Scalar Slots with Date/Time values on **Data Objects.**

2.1 NON-Requirements

The following potential requirements are **NOT CURRENTLY actual requirements,** and will be excluded from the current development.

1. Ability to Import and Export Date/Time Slot values (both SimObj Import/Export and via DMI operations).
2. Flexible configurable display format options. (It would be sufficient to provide a limited set of choices).

A single standard display format will be supported for each absolute or partial date variant.

3.0 Internal Representation

3.1 Definition: Full and Partial Date/Time Scope

Date/Time "Scope" refers the set of Date/Time "parts" (or "components") which make up a Date/Time. Full Date/Times have all parts defined. Partial Date/Times have only certain parts defined. The parts are:

1. Year
2. Month
3. Day
4. Hour
5. Minute
6. Second

Note about seconds: Although there is internal support for DateTime resolution down to seconds, the GUI components currently support only resolution down to minutes.

All Date/Time Scopes must have a valid RPL Symbolic Date/Time representation. The current development supports these Date/Time Scopes:

```
namespace PartialDateTime
{
   typedef enum { //--- Scope -------------------------
   Scope_Absolute    = 0,  // absolute date/time
   Scope_Month       = 1,  // e.g. February
   Scope_MonthAndDay = 2,  // e.g. January 2
   Scope_TimeOfYear  = 3,  // e.g. March 3, 9:25
   Scope_DayOfMonth  = 4,  // e.g. 27th
   Scope_TimeOfMonth = 5,  // e.g. 17th, 8:12
   Scope_Time        = 6,  // e.g. 16:20
   Scope_Year        = 7   // e.g. 2007
   } Scope;
...
}
```

Any single Slot (or AggSeries Slot Column or TableSlot Column) having the new DATETYPE unit type will support only one particular Partial DateTime Scope, (see note below). Instead of adding a new Partial DateTime Scope property to the Slot configuration, the particular Scope of a Slot will be represented as the Slot's User Unit. The following DATETIME user units are defined:

PartialDateTime::Scope	**DATETIME User Unit**	**Parts**
Scope_Absolute	"FullDateTime"	Year, Month, Day, Hour, Minute, Second
Scope_Month	"Month"	Month
Scope_MonthAndDay	"MonthAndDay"	Month, Day
Scope_TimeOfYear	"TimeOfYear"	Month, Day, Hour, Minute, Second
Scope_DayOfMonth	"DayOfMonth"	Day
Scope_TimeOfMonth	"TimeOfMonth"	Day, Hour, Minute, Second
Scope_Time	"TimeOfDay"	Hour, Minute, Second
Scope_Year	"Year"	Year

Note: The Partial DateTime Scope of Slot Values is **represented redundantly** by both the Unit configuration for the Slot (or Slot Column) AND the fractional part of the double-precision encoded DateTime values (see next section). The latter is needed for the interface to RPL Symbolic DateTimes since we are not extending the RPL type system for these values. In the Open Slot dialogs, any Slot cell value having a fractional part not consistent with the Slot (or Slot Column's) Partial DateTime Scope is shown with a yellow cell background.

3.2 "Double" (floating point value) Encoding of Date/Times

Conveniently, the integer subset of a "double" (double precision floating point number) far exceeds the number of seconds within the required time range -- currently 2000 years starting at year 1800. (The IEEE 754 specification for doubles includes a 52-bit mantissa, capable of representing all the seconds within 142 million years ... IF the full mantissa were to be used for a number of seconds).

The RiverWare **Date_Time** class, for representing absolute times, in seconds, from the beginning of the valid time range (currently starting at the beginning of the year 1800), uses an int64 (64 bit integer, typedef **seconds_t**) for that quantity.

To keep the RPL interface simple, a single common internal "double encoded" representation will be implemented, common to both Absolute Date/Times and Partial Date/Times. (This is a change from the early preliminary design).

The RPL "NumberToDate" and "DateToNumber" predefined functions need to operate on all forms of Absolute and Partial Date/Times without reference to auxiliary "type" information.

> Note: An *alternative* would be to augment the RPL type system to distinguish between the different forms of Date/Time values. This would basically double the number of supported RPL types. Also, currently the Assignment operator functions with only NUMERIC values, so that would have to be enhanced.

The "double encoding" of DateTimes will use:

- the **integer** (whole number) part of the double for the number of seconds since the beginning of the supported date range (currently, the beginning of year 1800).
- the **fractional** part for an indication of which date/time parts are relevant. Six binary fractional digits will be used to independently indicate which date/time parts are present. **Note:** in order for the **fractionless** repre-

sentation to have a useful meaning, **the sense of the factional bits will be negative:** all zeros will indicate that all date/time parts are significant, i.e. for an Absolute Date/Time.

Using 6 fractional binary digits for “scope” information effectively leaves 46 bits of mantissa for the integer part of the double-precision floating point number for the number of seconds. That’s 70,368,744,177,664 seconds, or about 2,231,000 years.

The 6 fractional binary digits will be assigned, in order of significance to: Year, Month, Day, Hour, Minute, Second. This table shows the binary fractions used for each part:

Date/Time Part	Binary Fraction	Decimal (positive sense)
Year	1 / 2	0.5
Month	1 / 4	0.25
Day	1 / 8	0.125
Hour	1 / 16	0.0625
Minute	1 / 32	0.03125
Second	1 / 64	0.015625

In **decimal representations** (e.g. in RiverWare model files), to accurately represent 6 binary fractional digits requires 6 decimal fractional digits. (It just works out that way). Standard RiverWare fractional “Precision” is not generally sufficient -- double-encoded date/time values will require “Extended (fractional) Precision”. The fraction values for the supported Partial Date/Time scopes are shown in the following table. (Note that the sense of the binary digits is negative).

PartialDateTime::Scope	**DATETIME User Unit**	**Parts**	**double-encoded fraction**
Scope_Absolute	“FullDateTime”	Year, Month, Day, Hour, Min, Sec.	0.000000
Scope_Month	“Month”	Month	0.734375
Scope_MonthAndDay	“MonthAndDay”	Month, Day	0.609375
Scope_TimeOfYear	“TimeOfYear”	Month, Day, Hour, Min, Sec	0.500000
Scope_DayOfMonth	“DayOfMonth”	Day	0.859375
Scope_TimeOfMonth	“TimeOfMonth”	Day, Hour, Min, Sec	0.750000
Scope_Time	“TimeOfDay”	Hour, Minute, Second	0.875000
Scope_Year	“Year”	Year	0.484375

--- (end) ---