

RiverWare Series Display Compression

Proposal for Persistence of Series Display Compression on Slots (only).

Author: Phil Weinstein, CADSWES
Edit Date: 10-12-2007

Contents

- [Overview](#)
- [Recommendation](#)
- [Reference: class Sim/DisplayCompressSpec](#)
- [Persistence Format](#)

Overview

The original [RiverWare Series Display Compression design document](#) recommended that Persistence not be implemented for the Series Display Compression feature.

That is, each time an Open SeriesSlot dialog, or any of the related Accounting Series display dialogs is opened, Series Display Compression would be OFF, and all compression parameters would have their **initial default value**:

Series Display Compression specification

Initial Default Settings:

Compression Mode:	Compress Repeated Values
Precision Type:	Display
Reference Value:	0.0

It was noted that if Persistence was to be implemented, the Display Compression Specification -- if active -- should be saved with each of the following entities:

1. Slots (for the Open Slot dialog)
2. Accounts (for the Edit Account dialog)
3. Object Account Summary Configurations (for the Object Account Summary dialog)
4. Exchanges (for the Exchange Balance dialog)

The nature of the use of this feature in these four applications **warrants reconsideration of an implementation of Series Display Compression settings persistence for only Slots.**

- Often Series Display Compression will be meaningful in the context of one particular Slot -- for example, as with [the first use case](#) provided by Shane Coors [USBR]. In that example, that Slot should always be displayed with Hidden Value compression with a Reference Value of 20.0.
- However, it is very unlikely that a "Series Display Compressed" presentation will be fundamental to any particular Account, Object Account Summary configuration, or Exchange. [Actually, of these three, the persistent compression settings could be useful for an Object Account Summary configuration, but this isn't yet certain].

Recommendation [Phil and Patrick, 10-10-2007]

Persistence of Series Display Compression settings should be implemented for Slots. That is, relevant **Slot records in RiverWare model files** (and Exported RiverWare Object files) should include a Display Compression Specification (class Sim/DisplayCompressSpec) if and only if compression was active the last time the Slot was viewed in an Open Slot dialog.

This means that the following classes should include a Display Compression Specification field. Probably this should be via a pointer to a dynamically allocated DisplayCompressSpec instance.

- SeriesSlot
- TableSeriesSlot

[Currently Display Compression Specification records currently live only in GUI components].

Reference: class Sim/DisplayCompressSpec

The DisplayCompressSpec currently includes the following fields.

```
class DisplayCompressSpec
{
private:
    bool _panelShown;
    bool _on;
    DisplayCompress::Op _op;
    DisplayCompress::Precision _prec;
    double _refValue;
    ...
};
```

DisplayCompress Definitions used above:

```
typedef enum {
    DISP_COMPR_OP_UND = 0,
    DISP_COMPR_OP_HIDE_DUPS, // Compress duplicate timestep rows.
    DISP_COMPR_OP_HIDE_NANS, // Hide rows having only NaNs.
    DISP_COMPR_OP_HIDE_VALUE, // Hide rows having only the Reference Value.
    DISP_COMPR_OP_HIDE_NAN_VAL // Hide rows having only NaNs or the Ref Value.
} Op; // Operation Code
```

```
typedef enum {
    DISP_COMPR_PREC_UND = 0,
    DISP_COMPR_PREC_DISPLAY, // Compare values using Display Precision
    DISP_COMPR_PREC_CONVERGENCE, // Compare values using the Slot's Convergence
    DISP_COMPR_PREC_EXACT // Compare values using exact equality
} Precision; // Precision Code
```

The following terse string representations are intended for persistence of these enumerated types.

```
QString DisplayCompress::opStrTerse (DisplayCompress::Op operation)
{
    switch (operation)
    {
        case DISP_COMPR_OP_UND: return ("OP_UND");
        case DISP_COMPR_OP_HIDE_DUPS: return ("OP_HIDE_DUPS");
        case DISP_COMPR_OP_HIDE_NANS: return ("OP_HIDE_NANS");
        case DISP_COMPR_OP_HIDE_VALUE: return ("OP_HIDE_VAL");
        case DISP_COMPR_OP_HIDE_NAN_VAL: return ("OP_HIDE_NAN_VAL");
    }
}
```

```

}

return ("OP_???");
}

```

```

QString DisplayCompress::precisionStrTerse (DisplayCompress::Precision prec)
{
    switch (prec)
    {
        case DISP_COMPR_PREC_UND:           return ("PREC_UND");
        case DISP_COMPR_PREC_DISPLAY:       return ("PREC_DISP");
        case DISP_COMPR_PREC_CONVERGENCE:   return ("PREC_CONV");
        case DISP_COMPR_PREC_EXACT:         return ("PREC_EXACT");
    }

    return ("PREC_???");
}

```

Persistence Format

A Tcl-parsable DisplayCompressSpec record could look like this:

```
"$s" displayCompress {shown} {on} {OP_HIDE_VAL} {PREC_DISP} 20.0
```

where:

- "\$s" is the Tcl variable for the current Slot
- {shown} could also be {hidden}
- {on} could also be {off}
- {OP_HIDE_VAL} could also be {OP_UND} or {OP_HIDE_DUPS} or {OP_HIDE_NANS} or {OP_HIDE_VAL} or {OP_HIDE_NAN_VAL}
- {PREC_DISP} could also be {PREC_UND} or {PREC_CONV} or {PREC_EXACT}

--- (end) ---