TValue5WebService User's Guide

1 CONTENTS

2 BACKGROUND

The TValue5WebService is designed to give access to all the calculation capabilities that the TValue™ desktop program provides. It is a modern web service created with the Microsoft Windows Communication Foundation architecture (WCF). Many modern programming languages will automatically create the bindings and data classes needed to use the TValue5WebService simply by adding a reference to the service. The service is typically licensed for use on an annual basis. Please contact TimeValue Software at 1-800-426-4741 to discuss licensing this service. For a tutorial on creating a WCF client, please see How to: Create a Windows Communication Foundation Client (https://msdn.microsoft.com/en-us/library/ms733133(v=vs.110).aspx).

3 SERVICE LOCATION

The service endpoint is https://www.Tvalue5WebService.com/TValue5Service.svc. The service is hosted on Amazon Web Servers. These servers are highly redundant and secure, with expected up times of 99.99% or better.

4 USING THE SERVICE

Most of the TValue functionality is achieved in a single web method called "Calculate". Calling Calculate with the appropriate parameters will allow you to solve any problem that the TValue desktop program can solve**. It is important that you learn how to use the TValue desktop program before you try to use the service**. The service has just 4 methods:

1. Calculate – Solves for any TValue problem.

2. GetLicenseInfo – Tests if your license key is valid.

3. GetBalance – Returns the principal and interest due on a particular date.

4. GetRates – Converts between the 4 rates types (Nominal, Effective, Periodic, Daily). Given one rate type it returns the other three.

5 CALCULATE METHOD

The calculate method will take an input cashFlowDataXml string and return an XML string representation of the answer. It will also produce an XML representation of the amortization schedule if the `createAmortizationSchedule` parameter is true.

```
string Calculate(string licenseKey,
        string cashFlowDataXml,
        TVRoundingType applyRoundingTo,
        TVEventType roundingEventType,
        int roundingSpecificLine,
        TVSpecificLineAdjust roundingSpecificLineAdjust,
        bool createAmortizationSchedule);
```

string licenseKey – This is the unique key supplied to each licensed user of the service.

string cashFlowDataXml – This XML string contains all the cash flow matrix data and events for the problem. See the section below entitled "CashFlowDataXML" for a description.

TVRoundingType applyRoundingTo – This parameter tells how to apply any leftover rounding. Possible values are:

```
TVRoundingLastPayment = 0,
TVRoundingFirstLoan = 1,
TVRoundingBallonAddNewPayment = 2,
TVRoundingSpecificLine = 3,
TVRoundingOpenBalance = 4,
TVRoundingLastInterestAmountIgnore = 5,
TVRoundingSystemDefault = 6
```

TVEventType roundingEventType – This parameter tells the event type to apply the rounding to. Possible values are:

```
TVLoanEvent = 8,
TVPaymentEvent = 9,
TVDepositEvent = 10,
TVWithdrawalEvent = 11,
TVUserLoanEvent1 = 12,
TVUserPaymentEvent1 = 13,
TVUserDepositEvent1 = 14,
TVUserWithdrawalEvent1 = 15,
```

TVUserLoanEvent2 = 16,
TVUserLoanEvent3 = 17,
TVUserPaymentEvent2 = 18,
TVUserPaymentEvent3 = 19,
TVUserDepositEvent2 = 20,
TVUserDepositEvent3 = 21,
TVUserWithdrawalEvent2 = 22,
TVUserWithdrawalEvent3 = 23

Int roundingSpecificLine – This parameter tells the line number of the event to apply the rounding to. This is just an integer value.

TVSpecificLineAdjust roundingSpecificLineAdjust – This parameter tells how to adjust the selected roundingSpecificLine. Possible values are:

TVSpecificLineAdjustAllAmounts = 0,
TVSpecificLineAdjustFirstAmount = 1,
TVSpecificLineAdjustLastAmount = 2,

bool createAmortizationSchedule – This parameter tells if an amortization schedule should be generated and returned as part of the XML return string.

6 CashFlowDataXml

The cashFlowDatXml string describes the TValue problem. It contains the Label, Nominal Annual Rate, and Compounding for the problem. It then lists all of the event lines. The TValue 5 desktop program uses this same XML representation when storing its files. You could open a TValue 5 file, and pass its contents in as the cashFlowDataXml value to the calculate method. **You will need to understand how to solve your problem with the TValue 5 desktop program before you can effectively use this service.**

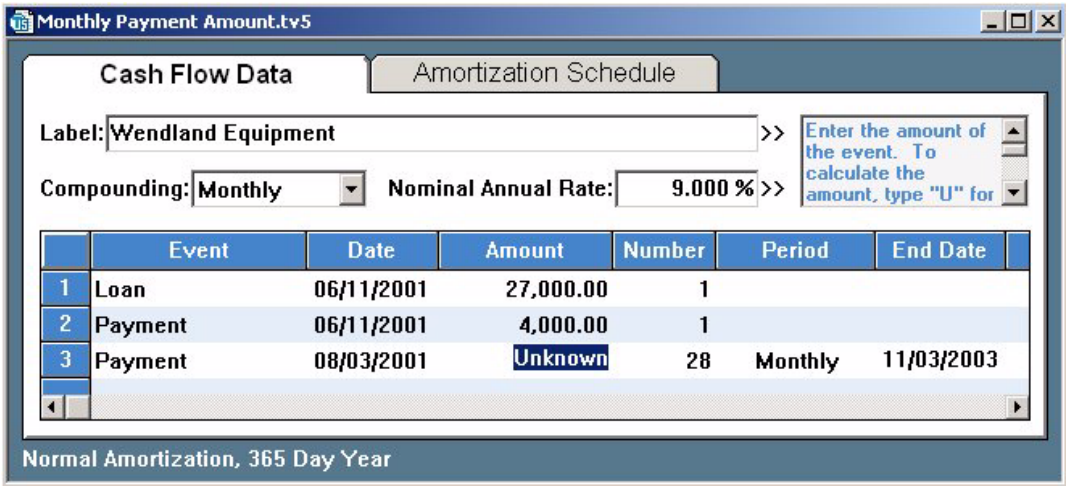
Let's take a look at The XML for a simple problem.

Facts Miller Equipment Co. sells machinery to Wendland on June 11, 2001 for $27,000. They allow a trade-in of $4000 on used equipment and take a 9 percent note calling for 28 equal monthly payments beginning on August 3, 2001.

Needed The monthly payment amount.

Settings This example assumes Normal amortization, 365 day year.

Here is the problem in TValue for Windows.



Note You could have subtracted the $4000 trade-in from the initial loan amount to arrive at a $23,000 loan amount. You then could have omitted the payment on line 2.

Solution The monthly payment will be $918.72.

The following is the XML for this problem.

```
<TValueCfm>
  <TValueVersion>5.1.0</TValueVersion>
  <CFMLabel>Wendland Equipment</CFMLabel>
  <ComputeMethod>0</ComputeMethod>
  <Compounding>6</Compounding>
  <NominalAnnualRate>0.09</NominalAnnualRate>
  <YearLength>365</YearLength>
  <Event>
    <EventType>8</EventType>
    <EventDate>06/01/2001</EventDate>
    <EventAmount>270000000</EventAmount>
    EventNumber>1</EventNumber>
  </Event>
  <Event>
    <EventType>9</EventType>
    <EventDate>06/01/2001</EventDate>
    <EventAmount>40000000</EventAmount>
    <EventNumber>1</EventNumber>
  </Event>
  <Event>
    <EventType>9</EventType>
    <EventDate>08/03/2001</EventDate>
    <EventAmount></EventAmount>
    <EventNumber>28</EventNumber>
    <EventPeriod>6</EventPeriod>
  </Event>
</TValueCfm>
```

A saved TValue Version 5 file will actually contain much more than what is above because of all the user settings and report options that it stores, but at the bottom you will see events listed just as shown above.

Let's go over each of the XML lines.

This is the start-tag for the **TValueCfm** element. The sub-tags within the TValueCfm element completely define the problem to be solved. The acronym **Cfm** stands for Cash Flow Matrix, this is synonymous with Cash Flow Data.

```
<TValueCfm>
```

The **TValueVersion** element specifies the version of TValue. This tag is required and should be "5.1.0".

```
<TValueVersion>5.1.0</TValueVersion>
```

The **CFMLabel** element specifies the label for the cash flow matrix.

```
<CFMLabel>Wendland Equipment</CFMLabel>
```

The **ComputeMethod** element should be one of the following values:

```
TVNormalAmortization      = 0,
TVRule78Amortization      = 1,
TVUSRuleAmortization      = 2,
TVCanadianAmortization    = 3
```

```
<ComputeMethod>0</ComputeMethod>
```

The **Compounding** element should be one of the following values:

```
TVContinuousCompound      = 0,
TVExactDaysCompound       = 1,
TVDailyCompound           = 2,
TVWeeklyCompound          = 3,
TVBiWeeklyCompound        = 4,
TVHalfMonthCompound       = 5,
TVMonthlyCompound         = 6,
TVTwoMonthCompound        = 7,
TVQuarterlyCompound       = 8,
TVFourMonthCompound       = 9,
TVFourWeekCompound        = 10,
TVSemiannualCompound      = 11,
TVAnnualCompound          = 12,
TVNoCompound              = 13
```

```
<Compounding>6</Compounding>
```

The **NominalAnnualRate** element is a decimal number.

```
<NominalAnnualRate>0.09</NominalAnnualRate>
```

The **YearLength** element must be 360, 364, or 365

```
<YearLength>365</YearLength>
```

The **Event** start-tag marks the beginning of the data for a single event in the cash flow matrix.

```
<Event>
```

The **EventType** element must be one on the following values:

TVRateChangeEvent = 4,
TVLoanEvent = 8,
TVPaymentEvent = 9,
TVDepositEvent = 10,
TVWithdrawalEvent = 11,
TVUserLoanEvent1 = 12,
TVUserLoanEvent2 = 16,
TVUserLoanEvent3 = 17,
TVUserPaymentEvent1 = 13,
TVUserPaymentEvent2 = 18,
TVUserPaymentEvent3 = 19,
TVUserDepositEvent1 = 14,
TVUserDepositEvent2 = 20,
TVUserDepositEvent3 = 21,
TVUserWithdrawalEvent1 = 15,
TVUserWithdrawalEvent2 = 22,
TVUserWithdrawalEvent3 = 23

```
<EventType>8</EventType>
```

The **EventDate** element must be in the format MM/DD/YYYY, where MM = the month, DD = the day and YYYY is the year.

```
<EventDate>06/01/2001</EventDate>
```

The **EventAmount** element is the Currency value. For amounts in dollars you take the dollar amount and multiple by 10,000 to get the Currency value. Thus $27,000 = 270000000.

```
<EventAmount>270000000</EventAmount>
```

The **EventNumber** element is an integer value between 1 and 999.

```
EventNumber>1</EventNumber>
```

We mark the end of the data for the event with the **</Event>** end-tag.

```
</Event>
```

We then add the second event with the following XML.

```
<Event>
  <EventType>9</EventType>
  <EventDate>06/01/2001</EventDate>
  <EventAmount>40000000</EventAmount>
  <EventNumber>1</EventNumber>
</Event>
```

The following is the XML for the third event. **Note that the EventAmount element has no value set. This is how you indicate that it is an unknown value.**

```
<Event>
  <EventType>9</EventType>
  <EventDate>08/03/2001</EventDate>
  <EventAmount></EventAmount>
  <EventNumber>28</EventNumber>
  <EventPeriod>6</EventPeriod>
</Event>
```

We finish the problem by inserting the **</TValueCfm>** end-tag.

```
</TValueCfm>
```

The Calculate method will take the above XML and solve for the unknowns, (in this case the unknown event amount in event 3).

7 AMORTIZATION SCHEDULE XML

If the parameter **createAmortizationSchedule** is True, then the Calculate method will add amortization XML to its return string.

The following is a sample of what the amortization XML will look like.

```
<TValueAmortizationSchedule>
  <Rounding>0</Rounding>
  <Compounding>6</Compounding>
  <NominalAnnualRate>0.09</NominalAnnualRate>
  <APR>0.0899946</APR>
  <FinanceCharge>27898900</FinanceCharge>
  <AmountFinanced>270000000</AmountFinanced>
  <TotalOfPayments>297898900</TotalOfPayments>
  <AmortizationLine>
            <AmortizationLineType>8</AmortizationLineType>
```

```
            <Date>06/01/2001</Date>
            <Loan1Amount>270000000</Loan1Amount>
            <Loan2Amount></Loan2Amount>
            <Loan3Amount></Loan3Amount>
            <Payment1Amount></Payment1Amount>
            <Payment2Amount></Payment2Amount>
            <Payment3Amount></Payment3Amount>
            <InterestAccrued>0</InterestAccrued>
            <InterestPaid>0</InterestPaid>
            <PrincipalPaid>0</PrincipalPaid>
            <UnpaidInterestBalance>0</UnpaidInterestBalance>
            <PrincipalBalance>270000000</PrincipalBalance>
            <TotalBalance>270000000</TotalBalance>
            <RateChangeRate></RateChangeRate>
            <RateChangeCompounding>13</RateChangeCompounding>
  </AmortizationLine>
  .
  .
  .
</TValueAmortizationSchedule>
```

Let's go over each of the lines in this amortization schedule XML.

This is the start-tag for the **TValueAmortizationSchedule** element. The report options affect what is listed in the amortization schedule, and thus the sub-elements within the TValueAmortizationSchedule element. This example shows the default sub-elements.

```
<TValueAmortizationSchedule>
```

The **Rounding** element tells the rounding amount as a currency value.

```
  <Rounding>0</Rounding>
```

The **Compounding** element tells the compounding that was in use. It is one of the following values:

```
TVContinuousCompound   = 0,
TVExactDaysCompound    = 1,
TVDailyCompound        = 2,
TVWeeklyCompound       = 3,
TVBiWeeklyCompound     = 4,
TVHalfMonthCompound    = 5,
TVMonthlyCompound      = 6,
TVTwoMonthCompound     = 7,
TVQuarterlyCompound    = 8,
TVFourMonthCompound    = 9,
TVFourWeekCompound     = 10,
TVSemiannualCompound   = 11,
TVAnnualCompound       = 12,
TVNoCompound           = 13
```

```
  <Compounding>6</Compounding>
```

The NominalAnnualRate element tells the rate used for the problem.

```
<NominalAnnualRate>0.09</NominalAnnualRate>
```

The APR information as well as the associated regulation "Z" disclosure amounts (Finance Charge, Amount Financed, and Total of Payments) are listed.

```
<APR>0.0899946</APR>
<FinanceCharge>27898900</FinanceCharge>
<AmountFinanced>270000000</AmountFinanced>
<TotalOfPayments>297898900</TotalOfPayments>
```

The **AmortizationLine** element lists all the possible columns that an amortization line might contain. Note that many of the amounts are blank (such as "**<Loan2Amount></Loan2Amount>**" in this example). This indicates that the value is not used in this amortization schedule.

The **AmortizationLineType** element lists the type of amortization line. It will be one of the following:

TVRateChangeLine = 4,
TVLoanLine = 8,
TVPaymentLine = 9,
TVDepositLine = 10,
TVWithdrawalLine = 11,
TVUserLoan1Line = 12,
TVUserLoan2Line = 16,
TVUserLoan3Line = 17,
TVUserPayment1Line = 13,
TVUserPayment2Line = 18,
TVUserPayment3Line = 19,
TVUserDeposit1Line = 14,
TVUserDeposit2Line = 20,
TVUserDeposit3Line = 21,
TVUserWithdrawal1Line = 15,
TVUserWithdrawal2Line = 22,
TVUserWithdrawal3Line = 23,
TVMonthTotalLine = 49,
TVQuarterTotalLine = 50,
TVAnnualTotalLine = 51,
TVGrandTotalLine = 52

```
<AmortizationLine>
        <AmortizationLineType>8</AmortizationLineType>
        <Date>06/01/2001</Date>
        <SequenceNumber>1</SequenceNumber>
        <Loan1Amount>270000000</Loan1Amount>
        <Loan2Amount></Loan2Amount>
        <Loan3Amount></Loan3Amount>
        <Payment1Amount></Payment1Amount>
        <Payment2Amount></Payment2Amount>
        <Payment3Amount></Payment3Amount>
        <InterestAccrued>0</InterestAccrued>
```

```
            <InterestPaid>0</InterestPaid>
            <PrincipalPaid>0</PrincipalPaid>
            <UnpaidInterestBalance>0</UnpaidInterestBalance>
            <PrincipalBalance>270000000</PrincipalBalance>
            <TotalBalance>270000000</TotalBalance>
            <RateChangeRate></RateChangeRate>
            <RateChangeCompounding>13</RateChangeCompounding>
  </AmortizationLine>
```

There will then be numerous **AmortizationLine** elements followed by the final **</TValueAmortizationSchedule>** end-tag.

```
</TValueAmortizationSchedule>
```

8 GetLicenseInfo Method

The GetLicenseInfo method simply takes a licenseKey string as an input and returns the name of the company associated with the key, or an error message.

```
string GetLicenseInfo(string licenseKey);
```

9 GetBalance Method

The GetBalance method is used to determine the balance on a particular date. It takes the same basic parameters as the Calculate method with the addition of the balance date desired. Please see the Calculate Method section of this document for a description of these parameters. It returns a BalanceInfo object which tells the amount on that date, and the interest and principle amounts for that date.

```
BalanceInfo GetBalance(string licenseKey,
   DateTime balanceDate,
   string cashFlowDataXml,
   TVRoundingType applyRoundingTo,
   TVEventType roundingEventType,
   int roundingSpecificLine,
   TVSpecificLineAdjust roundingSpecificLineAdjust);


public class BalanceInfo
{
   public int StatusCode;
   public string StatusMessage;
   public DateTime BalanceDate;
   public decimal Amount;
   public decimal Principal;
   public decimal Interest;
}
```

10 GetRates Method

The GetRates method is used to convert between the 4 rates types (Nominal, Effective, Periodic, Daily). Given one rate type it returns the other three.

```
Rates GetRates(string licenseKey, double rate, RateType rateType, TVCompoundPeriod compounding);
```

```
public class Rates
{
   public int StatusCode;
   public string StatusMessage;
   public double Nominal;
   public double Effective;
   public double Periodic;
   public double Daily;
}
```

```
public enum RateType
{
   Nominal,
   Effective,
   Periodic,
   Daily
}
```

```
public enum TVCompoundPeriod
{
   TVContinuousCompound = 0,
   TVExactDaysCompound = 1,
   TVDailyCompound = 2,
   TVWeeklyCompound = 3,
   TVBiWeeklyCompound = 4,
   TVHalfMonthCompound = 5,
   TVMonthlyCompound = 6,
   TVTwoMonthCompound = 7,
   TVQuarterlyCompound = 8,
   TVFourMonthCompound = 9,
   TVFourWeekCompound = 10,
   TVSemiannualCompound = 11,
   TVAnnualCompound = 12,
   TVNoCompound = 13,
}
```

For example, if you would like to convert an Effective rate of 5.875% with monthly compounding to the Nominal, Periodic, and Daily rates, make the following call:

Rates myRates = GetRates("myKey", 0.05875, RateType.Effective, TVCompoundPeriod.TVMonthlyCompound);

On return, myRates will contain:

```
{
   StatusCode: 0,
   StatusMessage: "Ok.",
   Nominal: 0.05722498051244429,
   Effective: 0.05875,
   Periodic: 0.0047687483760370242,
   Daily: 0.00015678076852724464
}
```

11 Status Codes

Most of the methods for this service can return an XML StatusCode, or contain a StatusCode field. Below is a list of the possible status codes.

```
// Status codes.
#define TVERR_Base       -10000000
#define TVERR_UserBase   TVERR_Base - 1000


// User Errors. Note: TVERR_UserBase = -10001000.
#define TVERR_DateLessThenMinimum                              (TVERR_UserBase - 0)
#define TVERR_DateGreaterThenMaximum                           (TVERR_UserBase - 1)
#define TVERR_EndDateLessThenMinimum                           (TVERR_UserBase - 2)
#define TVERR_EndDateGreaterThenMaximum                        (TVERR_UserBase - 3)
#define TVERR_CanNotCalculateNoUnknowns                        (TVERR_UserBase - 4)
#define TVERR_CanNotCalculateMoreThan1UnknownType              (TVERR_UserBase - 5)
#define TVERR_CanNotCalculateNotSolvable                       (TVERR_UserBase - 6)
#define TVERR_AmountTooSmall                                   (TVERR_UserBase - 7)
#define TVERR_AmountTooLarge                                   (TVERR_UserBase - 8)
#define TVERR_RateTooSmall                                     (TVERR_UserBase - 9)
#define TVERR_RateTooLarge                                     (TVERR_UserBase - 10)
#define TVERR_EventNumberTooSmall                              (TVERR_UserBase - 11)
#define TVERR_EventNumberTooLarge                              (TVERR_UserBase - 12)
#define TVERR_EventPeriodInvalidWithCurrentCompoundPeriod      (TVERR_UserBase - 13)
#define TVERR_CompoundPeriodInvalidWithCurrentEventPeriods     (TVERR_UserBase - 14)
#define TVERR_UnknownNominalAnnualRateAlreadySet               (TVERR_UserBase - 15)
#define TVERR_UnknownEventRateAlreadySet                       (TVERR_UserBase - 16)
#define TVERR_UnknownEventNumberAlreadySet                     (TVERR_UserBase - 17)
#define TVERR_UnknownEventAmountAlreadySet                     (TVERR_UserBase - 18)
#define TVERR_InvalidMethodCall                                (TVERR_UserBase - 19)
#define TVERR_InvalidEventType                                 (TVERR_UserBase - 20)
#define TVERR_InvalidEventTypeForTheCurrentCFM                 (TVERR_UserBase - 21)
#define TVERR_AmountUnknownWeightTooSmall                      (TVERR_UserBase - 23)
#define TVERR_AmountUnknownWeightTooLarge                      (TVERR_UserBase - 24)
#define TVERR_CFMNotEmptyCantSetDecimalPlaces                  (TVERR_UserBase - 25)
#define TVERR_DecimalPlacesTooSmall                            (TVERR_UserBase - 26)
#define TVERR_DecimalPlacesTooLarge                            (TVERR_UserBase - 27)
```

```
#define TVERR_InvalidNullParameter                              (TVERR_UserBase - 28)
#define TVERR_NoPaymentEventInCFM                               (TVERR_UserBase - 29)
#define TVERR_NoLoanEventInCFM                                  (TVERR_UserBase - 30)
#define TVERR_InvalidIndex                                      (TVERR_UserBase - 31)
#define TVERR_UserName1NotSet                                   (TVERR_UserBase - 32)
#define TVERR_UserName2NotSet                                   (TVERR_UserBase - 33)
#define TVERR_UserName3NotNull                                  (TVERR_UserBase - 34)
#define TVERR_UserName2NotNull                                  (TVERR_UserBase - 35)
#define TVERR_InvalidEmptyString                                (TVERR_UserBase - 36)
#define TVERR_UnknownEventName                                  (TVERR_UserBase - 37)
#define TVERR_Event0CanNotBeRateChange                          (TVERR_UserBase - 38)
#define TVERR_AfterSortEvent0WouldBeRateChange                  (TVERR_UserBase - 39)

#define TVERR_RateChangeEventDoesNotHaveAnAmount                (TVERR_UserBase - 41)
#define TVERR_RateChangeEventDoesNotHaveAnUnknownWeight         (TVERR_UserBase - 42)
#define TVERR_RateChangeEventDoesNotHaveAnEventNumber           (TVERR_UserBase - 43)
#define TVERR_RateChangeEventDoesNotHaveAnEventPeriod           (TVERR_UserBase - 44)
#define TVERR_ChangeWouldCauseInvalidEventPeriod                (TVERR_UserBase - 45)
#define TVERR_SpecificLineCanNotBeRateChangeEvent               (TVERR_UserBase - 46)
#define TVERR_RuleOf78DoesNotAllowOpenBalance                   (TVERR_UserBase - 47)
#define TVERR_RuleOf78RequiresFirstEventToBeALoan               (TVERR_UserBase - 48)
#define TVERR_RuleOf78DoesNotAllowMultipleLoans                 (TVERR_UserBase - 49)
#define TVERR_RuleOf78DoesNotAllowRateChangeEvents              (TVERR_UserBase - 50)
#define TVERR_RuleOf78DoesNotAllowSpecialSeriesEvents           (TVERR_UserBase - 51)
#define TVERR_RuleOf78OnlyAllowsLoanAndPaymentEvents            (TVERR_UserBase - 52)
#define TVERR_CanadianComputeDoesNotAllowExactDayCompounding    (TVERR_UserBase - 53)
#define TVERR_CanadianComputeDoesNotAllowContinuousCompounding  (TVERR_UserBase - 54)
#define TVERR_CanadianComputeDoesNotAllowPrincipalFirst         (TVERR_UserBase - 55)
#define TVERR_CanadianComputeDoesNotAllowNoCompoundingRateChange (TVERR_UserBase - 56)
#define TVERR_UnknownRateNotAllowed                             (TVERR_UserBase - 57)
#define TVERR_UnknownAmountNotAllowed                           (TVERR_UserBase - 58)
#define TVERR_UnknownEventNumberNotAllowed                      (TVERR_UserBase - 59)
#define TVERR_InvalidMonth                                      (TVERR_UserBase - 60)
#define TVERR_InvalidDay                                        (TVERR_UserBase - 61)
#define TVERR_RoundingNotAllowedWithThisSpecialSeriesEvent      (TVERR_UserBase - 62)
#define TVERR_FirstOrAllRoundingNotAllowedWithAmountStep        (TVERR_UserBase - 63)
#define TVERR_FirstOrAllRoundingNotAllowedWithPercentStep       (TVERR_UserBase - 64)
#define TVERR_FirstOrAllRoundingNotAllowedWithMonthlySkip       (TVERR_UserBase - 65)
#define TVERR_PrincipalFirstSeriesOnlyAllowedWithUSRuleCompute  (TVERR_UserBase - 66)
#define TVERR_FirstEventIsNotALoan                              (TVERR_UserBase - 67)
#define TVERR_PointsFeesAndChargesGreaterThanLoanAmount         (TVERR_UserBase - 68)
#define TVERR_CanNotInsertBeforeLoanLineWithPointsAndFees       (TVERR_UserBase - 69)
#define TVERR_CouldNotReadVersion3or4File                       (TVERR_UserBase - 70)
#define TVERR_DateIsOutOfSequence                               (TVERR_UserBase - 71)
#define TVERR_Only1LoanCanHavePointsAndFees                     (TVERR_UserBase - 72)
#define TVERR_UnknownAmountNotAllowedWithExistingFixedSeries    (TVERR_UserBase - 73)
#define TVERR_UnknownEventNumberNotAllowedWithExistingFixedSeries (TVERR_UserBase - 74)
#define TVERR_CalculateRequiresAtLeastOneEvent                  (TVERR_UserBase - 75)
#define TVERR_CanNotSelectAnEventTypeWhoseEventNameIsNull       (TVERR_UserBase - 76)
#define TVERR_InvalidAmortizationLineNumber                     (TVERR_UserBase - 77)
#define TVERR_EventDatesOutOfOrder                              (TVERR_UserBase - 78)
#define TVERR_CanNotRestoreUnknownsWhenNewUnknownsExist         (TVERR_UserBase - 80)
#define TVERR_NoCashFlowLines                                   (TVERR_UserBase - 81)
#define TVERR_USRuleComputeDoesNotAllowDailyCompounding         (TVERR_UserBase - 82)
#define TVERR_USRuleComputeDoesNotAllowContinuousCompounding    (TVERR_UserBase - 83)
#define TVERR_CanadianComputeDoesNotAllowContinousCompounding   (TVERR_UserBase - 84)
#define TVERR_CouldNotOpenFile                                  (TVERR_UserBase - 85)
#define TVERR_RemainingBalanceTooLarge                          (TVERR_UserBase - 86)
#define TVERR_RemainingBalanceTooSmall                          (TVERR_UserBase - 87)
#define TVERR_EventNameInUseCanNotBeSetToNull                   (TVERR_UserBase - 88)
```

```
#define TVERR_SpecialSeriesNotAllowedWithLoansOrRateChange                (TVERR_UserBase - 89)
#define TVERR_PrePaidInterestDaysMakeEvent1DateInvalid                    (TVERR_UserBase - 90)
#define TVERR_MonthlySkipInvalidWithCurrentCompoundPeriod                 (TVERR_UserBase - 91)
#define TVERR_MonthlySkipSeriesRequiresMonthlyPeriod                      (TVERR_UserBase - 92)
#define TVERR_InvalidDatePastEndOfMonth                                   (TVERR_UserBase - 93)
#define TVERR_ExpandNotAllowedWhenSpecialSeriesHasUnknown                 (TVERR_UserBase - 94)
#define TVERR_SortWouldPutEventInMiddleOfSpecialSeriesWithUnknown         (TVERR_UserBase - 95)
#define TVERR_ExpandWouldCreateTooManyCashFlowLines                       (TVERR_UserBase - 96)
#define TVERR_CannotSwitchFromUSRuleWhenPrincipalFirstExists              (TVERR_UserBase - 97)
#define TVERR_InRule78NoRateChangeSpecialSeriesMultipleLoansOpenBalanceDepositsWithdrawals
        (TVERR_UserBase - 98)
#define TVERR_InUSRuleNoDepositsWithdrawalsDailyOrContinousCompoundNoCompoundRateChange
        (TVERR_UserBase - 99)
#define TVERR_InCanadianCompoundingNoDailyOrContinousCompoundOrNoCompoundRateChange
        (TVERR_UserBase - 100)
#define TVERR_CompoundingCanNotBeNoneExceptInRateChange                   (TVERR_UserBase - 101)
#define TVERR_ContinousCompoundNotAllowedWithNoCompoundRateChange         (TVERR_UserBase - 102)
#define TVERR_NegativeRateNotAllowedWithContinuousCompound                (TVERR_UserBase - 103)
#define TVERR_DailyExactDaysAndContinuousNotAllowedWithExistingFixedSeries (TVERR_UserBase - 104)
#define TVERR_LastEventCannotBeRateChange                                 (TVERR_UserBase - 105)
#define TVERR_FixedPrincipalAmountCanNotBeZeroUseInterestOnly             (TVERR_UserBase - 106)
#define TVERR_FileCouldNotBeOpened                                        (TVERR_UserBase - 107)
#define TVERR_FileAccessError                                             (TVERR_UserBase - 108)
#define TVERR_NotATValueFile                                              (TVERR_UserBase - 109)
#define TVERR_MallocFailed                                                (TVERR_UserBase - 110)
#define TVERR_FileWriteFailed                                             (TVERR_UserBase - 111)
#define TVERR_CompoundPeriodWouldMakeRateTooHigh                          (TVERR_UserBase - 112)
#define TVERR_BalanceDateIsBeforeStartDate                                (TVERR_UserBase - 113)
#define TVERR_DepositsCanNotHaveFixedPrincipalOrInterestOnlySeries        (TVERR_UserBase - 114)
#define TVERR_NumberBeforeChangeMustBeZeroOrGreater                       (TVERR_UserBase - 115)
#define TVERR_NumberBeforeChangeIsTooLarge                                (TVERR_UserBase - 116)
#define TVERR_NumberToMakeOrSkipMustBeZeroOrGreater                       (TVERR_UserBase - 117)
#define TVERR_NumberToMakeOrSkipIsTooLarge                                (TVERR_UserBase - 118)
#define TVERR_CannotCalculateBalanceAmount                                (TVERR_UserBase - 119)
#define TVERR_CannotCalculateBalanceDate                                  (TVERR_UserBase - 120)
#define TVERR_UnknownNumberNotAllowedWithLoansThatHavePointsAndFees       (TVERR_UserBase - 121)
#define TVERR_SortWouldCreateTooManyCashFlowLines                         (TVERR_UserBase - 122)
#define TVERR_InsertWouldExceedTheMaximumNumberOfCashFlowLines            (TVERR_UserBase - 123)
#define TVERR_DuplicateEventName                                          (TVERR_UserBase - 124)
#define TVERR_InvalidVariantType                                          (TVERR_UserBase - 125)
#define TVERR_NoDateEntered                                               (TVERR_UserBase - 126)
#define TVERR_DailyCompoundNotAllowedWithCanadianOddDaysStraightLine      (TVERR_UserBase - 127)
#define TVERR_UserCanceled                                                (TVERR_UserBase - 128)
#define TVERR_CalculateWouldProduceNegativeRateWithContinuousCompounding  (TVERR_UserBase - 129)
#define TVERR_CalculateWouldProduceAnAmountThatIsTooLarge                 (TVERR_UserBase - 130)
#define TVERR_AfterSortLoanWithPointsAndFeesWouldNoLongerBeEvent0         (TVERR_UserBase - 131)
#define TVERR_Version4CannotHaveMoreThan400CashFlowLines                  (TVERR_UserBase - 132)
#define TVERR_PastDllExpirationDate                                       (TVERR_UserBase - 133)
#define TVERR_CalculateWouldProduceARateThatIsTooLarge                    (TVERR_UserBase - 134)
#define TVERR_CalculateWouldProduceAnEventNumberThatIsTooLarge            (TVERR_UserBase - 135)
#define TVERR_CfmObjectHasBeenDeleted                                     (TVERR_UserBase - 136)
#define TVERR_InvalidCompoundPeriod                                       (TVERR_UserBase - 137)
#define TVERR_InvalidEventPeriod                                          (TVERR_UserBase - 138)
#define TVERR_InvalidComputeMethod                                        (TVERR_UserBase - 139)
#define TVERR_InvalidYearLength                                           (TVERR_UserBase - 140)
#define TVERR_InvalidCanadianYearBasis                                    (TVERR_UserBase - 141)
#define TVERR_InvalidCanadianOddDays                                      (TVERR_UserBase - 142)
#define TVERR_InvalidRateMode                                             (TVERR_UserBase - 143)
```

```
// System Errors.
#define TVERR_UnableToCreateRate                        (TVERR_Base - 1)
#define TVERR_UnableToGetEffectiveRate                  (TVERR_Base - 2)
#define TVERR_UnableToGetPeriodicRate                   (TVERR_Base - 3)
#define TVERR_UnableToGetDailyRate                      (TVERR_Base - 4)
#define TVERR_UnableToGetNominalFromEffectiveRate       (TVERR_Base - 5)
#define TVERR_UnableToGetNominalFromPeriodicRate        (TVERR_Base - 6)
#define TVERR_UnableToGetNominalFromDailyRate           (TVERR_Base - 7)
#define TVERR_UnableToCreateCalculationCFM              (TVERR_Base - 8)
#define TVERR_UnableToDestroyCalculationCFM             (TVERR_Base - 9)
#define TVERR_UnableToSetYearLength                     (TVERR_Base - 10)
#define TVERR_UnableToSetAmortMethod                    (TVERR_Base - 11)
#define TVERR_UnableToSetCanadianBasis                  (TVERR_Base - 12)
#define TVERR_UnableToSetOpenBalanceFlag                (TVERR_Base - 13)
#define TVERR_UnableToGetMaxAmountEntryValue            (TVERR_Base - 14)
#define TVERR_Calculation_SetLine_Failed                (TVERR_Base - 20)
#define TVERR_Calculation_SetEvent_Failed               (TVERR_Base - 21)
#define TVERR_Calculation_SetDate_Failed                (TVERR_Base - 22)
#define TVERR_Calculation_SetAmountUnknown_Failed       (TVERR_Base - 23)
#define TVERR_Calculation_SetAmount_Failed              (TVERR_Base - 24)
#define TVERR_Calculation_SetUnknownNumber_Failed       (TVERR_Base - 25)
#define TVERR_Calculation_SetPaymentNumber_Failed       (TVERR_Base - 26)
#define TVERR_Calculation_SetPeriod_Failed              (TVERR_Base - 27)
#define TVERR_Calculation_RateCreate_Failed             (TVERR_Base - 28)
#define TVERR_Calculation_SetRate_Failed                (TVERR_Base - 29)
#define TVERR_Calculation_SetRateUnknown_Failed         (TVERR_Base - 30)
#define TVERR_Calculation_SetSeries_Failed              (TVERR_Base - 31)
#define TVERR_Calculation_SetSeriesData_Failed          (TVERR_Base - 32)
#define TVERR_Calculation_SetPIAType_Failed             (TVERR_Base - 33)
#define TVERR_AmortizeFailed                            (TVERR_Base - 40)
#define TVERR_AmortizeGetSizeFailed                     (TVERR_Base - 41)
#define TVERR_AmortizeSetLineFailed                     (TVERR_Base - 42)
#define TVERR_AmortizeGetDataFailed                     (TVERR_Base - 43)
#define TVERR_AmortizeSetFiscalYearFailed               (TVERR_Base - 44)

#define TVERR_Fire_NewEventAdded_Failed                 (TVERR_Base - 46)
#define TVERR_Fire_EventDeleted_Failed                  (TVERR_Base - 47)
#define TVERR_Fire_AllEventsDeleted_Failed              (TVERR_Base - 48)
#define TVERR_Fire_WasSorted_Failed                     (TVERR_Base - 49)
#define TVERR_Fire_WasExpanded_Failed                   (TVERR_Base - 50)
#define TVERR_Fire_WasCompressed_Failed                 (TVERR_Base - 51)
#define TVERR_Fire_LabelChanged_Failed                  (TVERR_Base - 52)
#define TVERR_Fire_CompoundPeriodChanged_Failed         (TVERR_Base - 53)
#define TVERR_Fire_NominalAnnualRateChanged_Failed      (TVERR_Base - 54)
#define TVERR_Fire_EventTypeChanged_Failed              (TVERR_Base - 55)
#define TVERR_Fire_EventDateChanged_Failed              (TVERR_Base - 56)
#define TVERR_Fire_EventAmountChanged_Failed            (TVERR_Base - 57)
#define TVERR_Fire_EventNumberChanged_Failed            (TVERR_Base - 58)
#define TVERR_Fire_EventPeriodChanged_Failed            (TVERR_Base - 59)
#define TVERR_Fire_EventEndDateChanged_Failed           (TVERR_Base - 60)
#define TVERR_Fire_EventNoteChanged_Failed              (TVERR_Base - 61)
#define TVERR_Fire_EventRateCompoundChanged_Failed      (TVERR_Base - 62)
#define TVERR_Fire_EventRateChanged_Failed              (TVERR_Base - 63)

#define TVERR_Calculation_GetLabels_Failed              (TVERR_Base - 75)
#define TVERR_Calculation_GetRate_Failed                (TVERR_Base - 76)
#define TVERR_Calculation_RateVP_Failed                 (TVERR_Base - 77)
#define TVERR_Calculation_GetEvent_Failed               (TVERR_Base - 78)
#define TVERR_Calculation_GetStartDate_Failed           (TVERR_Base - 79)
#define TVERR_Calculation_GetAmount_Failed              (TVERR_Base - 80)
```

```
#define TVERR_Calculation_GetPaymentNumber_Failed         (TVERR_Base - 81)
#define TVERR_Calculation_GetPeriod_Failed                (TVERR_Base - 82)
#define TVERR_Calculation_GetSeries_Failed                (TVERR_Base - 83)
#define TVERR_Calculation_GetUnknownField_Failed          (TVERR_Base - 84)
#define TVERR_Calculation_GetAmortMethod_Failed           (TVERR_Base - 85)
#define TVERR_Calculation_GetYearLength_Failed            (TVERR_Base - 86)
#define TVERR_Calculation_GetCanadianQuotation_Failed     (TVERR_Base - 87)
#define TVERR_Calculation_GetDailyMode_Failed             (TVERR_Base - 88)
#define TVERR_Calculation_UnknownWeight_Failed            (TVERR_Base - 89)
#define TVERR_Calculation_GetLoanDetail_Failed            (TVERR_Base - 90)
#define TVERR_Calculation_GetUnknownRow_Failed            (TVERR_Base - 91)
#define TVERR_Calculation_GetSeriesData_Failed            (TVERR_Base - 92)
#define TVERR_Calculation_GetUserEvents_Failed            (TVERR_Base - 93)
#define TVERR_Calculation_Compress_Failed                 (TVERR_Base - 94)
#define TVERR_Calculation_Expand_Failed                   (TVERR_Base - 95)
#define TVERR_Calculation_Sort_Failed                     (TVERR_Base - 96)
#define TVERR_Calculation_GetMaxLine_Failed               (TVERR_Base - 97)
#define TVERR_Calculation_SetAmountUnknownWeight_Failed   (TVERR_Base - 98)
#define TVERR_Calculation_GetUnknownPayment_Failed        (TVERR_Base - 99)
#define TVERR_Calculation_CompressLine_Failed             (TVERR_Base - 100)
#define TVERR_Calculation_ExpandLine_Failed               (TVERR_Base - 101)
#define TVERR_Calculation_SetEventTypeID_Failed           (TVERR_Base - 102)
#define TVERR_Calculation_GetEventTypeID_Failed           (TVERR_Base - 103)
#define TVERR_Calculation_ComputeAPR_Failed               (TVERR_Base - 104)
#define TVERR_Calculation_SetLoanDetail_Failed            (TVERR_Base - 105)
#define TVERR_Calculation_GetReportOptions_Failed         (TVERR_Base - 106)

#define TVERR_Calculation_GetFooter_Failed                (TVERR_Base - 108)
#define TVERR_Calculation_GetHeaders_Failed               (TVERR_Base - 109)
#define TVERR_Calculation_GetPIAType_Failed               (TVERR_Base - 110)
#define TVERR_Calculation_InvalidMonthlySkipSequence      (TVERR_Base - 111)
#define TVERR_Calculation_Insert_Failed                   (TVERR_Base - 112)
#define TVERR_Calculation_AddPeriod_Failed                (TVERR_Base - 113)
#define TVERR_Calculation_SetNumberUnknown_Failed         (TVERR_Base - 114)
#define TVERR_Calculation_RateSetEffective_Failed         (TVERR_Base - 115)
#define TVERR_Calculation_RateSetPeriodic_Failed          (TVERR_Base - 116)
#define TVERR_Calculation_RateSetDaily_Failed             (TVERR_Base - 117)
#define TVERR_Calculation_RateNominal_Failed              (TVERR_Base - 118)
#define TVERR_Calculation_WriteToFile_Failed              (TVERR_Base - 119)
#define TVERR_Calculation_SetLabels_Failed                (TVERR_Base - 120)
#define TVERR_Calculation_SetHeaders_Failed               (TVERR_Base - 121)
#define TVERR_Calculation_SetFooter_Failed                (TVERR_Base - 122)
#define TVERR_Calculation_SetReportOptions_Failed         (TVERR_Base - 123)
#define TVERR_Calculation_GetOpenFlag_Failed              (TVERR_Base - 124)

#define TVERR_NewFailed_OutOfRamMemory                    (TVERR_Base - 150)
#define TVERR_OldTValueDll                                (TVERR_Base - 151)
#define TVERR_UnableToSetRoundPrepaidInterestDailyAmount  (TVERR_Base - 152)
```