

## Technical Documentation for RUGs V5.12 DLLs DLL Version 1.11

Stepwise Systems, Inc.  
May 12, 1998

### Introduction

This documentation describes two DLLs which can be used to calculate RUG-III Version 5.12 from MDS 2.0 data. R512\_16.DLL is a 16-bit DLL which can be called from 16-bit Visual Basic 3.0 or 4.0 applications or from 16-bit C++ applications. R512\_32.DLL is a 32-bit ActiveX DLL which can be called from Visual Basic 5.0 or from 32-bit C++ applications.

Both DLLs accept an input string containing an MDS data record in the standard HCFA format and return the RUGs value and associated information. *Please note that the current versions of these DLLs (Version 1.11) support only the 44 group model; the 34-group RUGs model is not currently supported.*

These DLLs were developed by Stepwise Systems, Inc. of Austin, Texas under subcontract with Fu Associates, Ltd. through contract number GS35F-4052G with the Health Care Financing Administration. Red Bluff Systems of Austin, Texas provided invaluable assistance in developing the interface with C++.

The ZIP file containing the software and this documentation may be freely distributed and you may freely use and distribute the DLLs in your applications. Please refer to the license agreement at the end of this documentation for details.

There are four sections later in this document which describe the use of each of the two DLLs from Visual Basic and from C++. Following these sections, there are two sections which should be read by all users. The first describes the input and output parameters used when calling either DLL. The second section describes the case mix index array which is required by the DLLs.

### Contents of the ZIP File

R512\_111.ZIP contains the following files:

- **Readme.txt** A brief description of the DLLs and the ZIP file contents.
- **TechDoc.pdf** This documentation file in Adobe Acrobat® format. You must have the Adobe Acrobat® reader to view or print this file. The reader is available for free from <http://www.adobe.com> and from many other sites on the Internet.
- **R512\_16.dll** The 16-bit DLL.
- **R512\_32.dll** The 32-bit DLL.
- **VbDemo16.\*** Visual Basic 3.0 project files and an EXE demonstrating the use of the 16-bit DLL from Visual Basic.
- **VbDemo32.\*** Visual Basic 5.0 project files and an EXE demonstrating the use of the 32-bit DLL from Visual Basic.
- **CDemo16.\*** Microsoft Visual C++ Version 1.51 project files and an EXE demonstrating the use of the 16-bit DLL from C++.
- **Cdemo32.\*** Microsoft Visual C++ Version 5.0 project files and an EXE demonstrating the use of the 32-bit DLL from C++.
- **Test512a.asc** A file containing test MDS 2.0 data. This file is used by the four demo programs.

## Running the Demo Programs

The four demo programs all work similarly. Each reads MDS 2.0 data from TEST512A.ASC and writes an output file to RUGLIST.TXT. The output file will contain the calculated RUGs value, other output parameters from the DLL, and RUGs values obtained from the MDS record itself (for comparison with the calculated values).

Please refer to the notes below for information about where to place the DLL's on a target system and how to register them with Windows. In addition, please note that TEST512A.ASC must be in the application directory for the demo programs to run properly. RUGLIST.TXT will be written to the application directory. Also note that record #701 in RUGLIST.TXT shows an error code (-4) and a blank RUGs value (the error codes are explained later in this document). This error record is intentional and occurs because the corresponding input record contains an out-of-range reason for assessment code.

## Calling R512\_16.DLL from Visual Basic

R512\_16.DLL contains the function vbRugCalc44 which can be called from applications developed using Microsoft Visual Basic 3.0 or 4.0. A sample application called vbDemo16 is included with R512\_111.ZIP which illustrates the use of the DLL from a Visual Basic application.

Note that proper operation of the DLL requires that the following OLE2 files be available to the application:

- OLE2.DLL
- OLE2DISP.DLL
- OLE2NLS.DLL
- COMPOBJ.DLL
- STORAGE.DLL

If you distribute these files to client machines, make sure you distribute the versions which are appropriate for the target operating system.

In order to use R512\_16.DLL you should place it in your \windows\system directory. In addition, your application must have the following declaration (note that the underscore character below is used as a continuation character in this documentation, but such line continuations are not allowed in VB 3 or 4):

```
Declare Function vbRugCalc44 Lib "R512_16.DLL" (MdsRecord As String, _
    CalcType As String, QuarterlyFlag As Integer, CmiArray As Single, Rugs As String, _
    CmiValue As Single, AdlSum As Integer, CpsSum As Integer, RugsVersion As String, _
    DllVersion As String) As Integer
```

An example function call is as follows:

```
iRet = vbRugCalc44(sStringIn, "mcare", 0, aCmiA01(0), sRug, nCmiValue, _
    iAdlSum, iCpsSum, sRugsVersion, sDllVersion)
```

The case mix index (CMI) array (described below) should be dimensioned as follows:

```
Dim CmiArray() As Single
ReDim CmiArray(49)
```

## Calling R512\_16.DLL from C++

R512\_16.DLL contains the function cRugCalc44 which can be called from applications developed using Microsoft Visual C++ Version 1.51. A sample application called cDemo16 is included with R512\_111.ZIP which illustrates the use of the DLL from a C++ application.

Note that proper operation of the DLL requires that the following OLE2 files be available to the application:

- OLE2.DLL
- OLE2DISP.DLL
- OLE2NLS.DLL
- COMPOBJ.DLL
- STORAGE.DLL

If you distribute these files to client machines, make sure you distribute the versions which are appropriate for the target operating system.

In order to use R512\_16.DLL you should place it in your \windows\system directory.

## ***Calling R512\_32.DLL from Visual Basic***

R512\_32.DLL contains a class called Rug512 which can be called from applications developed using Microsoft Visual Basic 5.0. A sample application called vbDemo32 is included with R512\_111.ZIP which illustrates the use of the DLL from a Visual Basic application.

When you distribute an application which uses R512\_32.DLL, you must also distribute MSVBVM50.DLL which is Visual Basic 5.0's runtime DLL (this support file is required by all VB5 applications). You must also register R512\_32.DLL using RegSvr32.Exe.

In order to use the Rug512 class, you must create a local instance of it using a statement like the following:

```
Dim Rug As New Rug512
```

To perform the RUGs calculation, you must follow these three steps:

1. Set certain required properties.
2. Invoke the RugCalc method.
3. Get the values of output properties.

A later section of this documentation (*Parameters Used by the DLLs*) lists the parameters used to calculate RUGs. The parameters listed as input parameters are the ones which you must set before invoking the RugCalc method. The parameters listed as output parameters are the ones which you should get after invoking the RugCalc method.

The statements illustrate the usage of the Rug512 class:

```
'Set properties to prepare for calculation
Rug.MdsRecord = sStringIn    'Sets MDS record string
Rug.CalcType = "Mcare"      'Sets calculation type
Rug.QuarterlyFlag = 0       'Sets quarterly flag
Rug.CmiArray = aCmiA01()    'Sets case mix index array

'Invoke RugCalc method to perform calculation
Rug.RugCalc

'Get return value properties
sRug = Rug.Rugs              'Gets RUGs group
nCmiValue = Rug.CmiValue     'Gets CMI corresponding to RUGs group
iAdlSum = Rug.AdlSum         'Gets ADL scale score
iCpsSum = Rug.CpsSum         'Gets CPS scale score
sRugsVersion = Rug.RugsVersion 'Gets RUGs version code
sDllVersion = Rug.DllVersion 'Gets DLL version
```

```
iError = Rug.Error
```

```
'Gets calculation error code
```

## Calling R512\_32.DLL from C++

R512\_32.DLL contains a class called Rug512 which can be called from applications developed using Microsoft Visual C++ Version 5.0. A sample application called cDemo32 is included with R512\_111.ZIP which illustrates the use of the DLL from a C++ application.

When you distribute an application which uses R512\_32.DLL, you must also distribute MSVBVM50.DLL which is Visual Basic 5.0's runtime DLL (this support file is required since the DLL was developed using Visual Basic 5.0). You must also register R512\_32.DLL using RegSvr32.Exe.

In order to use the Rug512 class, you must first create a local instance of it. You may then perform the RUGs calculation in the following three steps:

1. Set certain required properties.
2. Invoke the RugCalc method.
3. Get the values of output properties.

A later section of this documentation (*Parameters Used by the DLLs*) lists the parameters used to calculate RUGs. The parameters listed as input parameters are the ones which you must set before invoking the RugCalc method. The parameters listed as output parameters are the ones which you should get after invoking the RugCalc method.

Please refer to the sample application (cDemo32) for details regarding instantiating the class, setting and getting properties, and invoking the RugCalc method.

## Parameters Used by the DLLs

Both of the DLLs use the same input and output parameters. Note that for the 32-bit DLL, the parameters correspond exactly with properties. You must “set” the properties listed as input parameters, and “get” the properties listed as output parameters.

| Parameter Name | Parameter Type | Data Type | Description   |
|----------------|----------------|-----------|---|
| MdsRecord      | Input          | String    | A string containing the MDS record in standard format (length=1812 bytes).  |
| CalcType       | Input          | String    | <p>A string containing one of the following three values designating the type of RUGs calculation to perform:</p> <ul style="list-style-type: none"> <li>• <i>Mcare</i> = special Medicare classification</li> <li>• <i>Hier</i> = hierarchical classification</li> <li>• <i>Index</i> = index maximizing classification</li> </ul> <p>Upper, lower, or mixed case is acceptable when supplying this parameter.</p> |

| Parameter Name | Parameter Type | Data Type              | Description   |
|----------------|----------------|------------------------|---|
| QuarterlyFlag  | Input          | Integer                | <p>The quarterly flag indicates whether RUGs is to be calculated for quarterly assessments.</p> <p>0 = RUGs not calculated for quarterlies</p> <p>1 = RUGs is calculated for quarterlies</p> <p>The function will automatically determine whether the MDS record corresponds to a quarterly assessment.</p>   |
| CmiArray       | Input          | Single precision array | <p>CmiArray is a single-precision array containing case mix indices (CMI's). This array affects the RUGs classification only if the CalcType is Mcare or Index, but it must always be supplied (you can initialize it with zeroes if the CalcType is Hier). The array must be dimensioned from 0 to 49. Put the 49 CMI indices in elements 1 to 49 (element 0 can be initialized to 0).</p> <p>If you are calling R512_16.DLL from Visual Basic, you must pass a reference to the first element in the array (e.g., CmiArray(0)) (see the example function call on page 2).</p> <p>The RUGs groups corresponding to the 49 elements in CmiArray are listed in a later section of this document.</p> |
| Rugs           | Output         | String                 | <p>The function will return a 3-byte string containing the RUGs group. When you call the function, you can initialize this variable to an empty string.</p>   |
| CmiValue       | Output         | Single precision       | <p>CmiValue is a single-precision variable which on return will contain the case mix index value (from CmiArray) corresponding to the resulting RUGs value.</p>   |
| AdlSum         | Output         | Integer                | <p>AdlSum is an integer variable which on return will contain the RUG-III ADL scale score.</p>  |
| CpsSum         | Output         | Integer                | <p>CpsSum is an integer variable which on return will contain the CPS (Cognitive Performance Scale) score.</p>  |
| RugsVersion    | Output         | String                 | <p>RugsVersion is a string variable which returns the version of the RUG-III classification logic which was used. This version is designated using the code specified in the RUG-III Version 5.12 specifications. For DLL Version 1.11, this value will always be "07", the designation for the 44 group RUG-III model of RUG-III Version 5.12. When you call the function, you can initialize this variable to an empty string.</p>  |
| DllVersion     | Output         | String                 | <p>DllVersion is a string which returns a value of "1.11", the version number for the present DLL. When you call the function, you can initialize this variable to an empty string.</p>   |

| Parameter Name | Parameter Type | Data Type | Description   |
|----------------|----------------|-----------|---|
| Error          | Output         | Integer   | <p>If you are using the 16-bit DLL, vbRugCalc or cRugCalc will return an integer error code. If you are using the 32-bit DLL, you can obtain the error code by getting the <i>Error</i> property. Either way, the error code will have one of the following values:</p> <ul style="list-style-type: none"> <li>0 = No DLL calling error, RUGs was calculated or set to the default value if out-of-range values were found for any RUGs item</li> <li>1 = CalcType parameter was invalid</li> <li>2 = CMI array was all zeroes when non-zero values are required (i.e., when CalcType is <i>Mcare</i> or <i>Index</i>)</li> <li>3 = QuarterlyFlag was invalid</li> <li>4 = RUGs not calculated for this type of assessment record. This code will always be returned for certain assessment types (e.g., a discharge record). It may be returned for a quarterly assessment depending upon the value of QuarterlyFlag. This error code will also be returned if the reasons for assessment codes (MDS items AA8a and AA8b) are invalid (e.g., out of range).</li> </ul> |

## Case Mix Index Array

As of this writing, four sets of case mix indices (CMIs) have been defined for RUGs-III Version 5.12:

- A01: Medicare PPS Rural -- 07/01/98
- A02: Medicare PPS Urban -- 07/01/98
- B01: Medicaid 34 group -- Nursing Only
- B02: Research 44 group -- Nursing only

Each of the example programs supplied with the DLLs contains four routines (LoadCmiA01, LoadCmiA02, LoadCmiB01, and LoadCmiB02) which will load the indices which correspond to each of these CMI sets into a single precision array.

Please note the following:

- B01 is reserved for future use with the 34-group model which is not supported by the current versions of the DLLs.
- B02 is not appropriate for use with Medicare or index maximizing calculation types.

The indices must be loaded into the CMI array in the following order:

| Element Number | RUG3 Group                       |
|----------------|----------------------------------|
| 0              | Required dummy element--set to 0 |

| <b>Element Number</b> | <b>RUG3 Group</b>                                    |
|-----------------------|--|
| 1                     | RUC: Rehabilitation Ultra High / ADL 16 – 18         |
| 2                     | RUB: Rehabilitation Ultra High / ADL 9 – 15          |
| 3                     | RUA: Rehabilitation Ultra High / ADL 4 - 8           |
| 4                     | RVC: Rehabilitation Very High / ADL 16 – 18          |
| 5                     | RVB: Rehabilitation Very High / ADL 9 – 15           |
| 6                     | RVA: Rehabilitation Very High / ADL 4 - 8            |
| 7                     | RHC: Rehabilitation High / ADL 13 – 18               |
| 8                     | RHB: Rehabilitation High / ADL 8 – 12                |
| 9                     | RHA: Rehabilitation High / ADL 4 - 7                 |
| 10                    | RMC: Rehabilitation Medium / ADL 15 – 18             |
| 11                    | RMB: Rehabilitation Medium / ADL 8 – 14              |
| 12                    | RMA: Rehabilitation Medium / ADL 4 - 7               |
| 13                    | RLB: Rehabilitation Low / ADL 14 – 18                |
| 14                    | RLA: Rehabilitation Low / ADL 4 – 13                 |
| 15                    | SE3: Extensive Special Care 3 / ADL > 6              |
| 16                    | SE2: Extensive Special Care 2 / ADL > 6              |
| 17                    | SE1: Extensive Special Care 1 / ADL > 6              |
| 18                    | SSC: Special Care / ADL 17 – 18                      |
| 19                    | SSB: Special Care / ADL 15 – 16                      |
| 20                    | SSA: Special Care / ADL 4 – 14                       |
| 21                    | CC2: Clin. Complex with Depression / ADL 17 - 18     |
| 22                    | CC1: Clinically Complex / ADL 17 – 18                |
| 23                    | CB2: Clin. Complex with Depression / ADL 12 - 16     |
| 24                    | CB1: Clinically Complex / ADL 12 – 16                |
| 25                    | CA2: Clin. Complex with Depression / ADL 4 - 11      |
| 26                    | CA1: Clinically Complex / ADL 4 – 11                 |
| 27                    | IB2: Cog. Impairment with Nursing Rehab / ADL 6 - 10 |
| 28                    | IB1: Cognitive Impairment / ADL 6 – 10               |
| 29                    | IA2: Cog. Impairment with Nursing Rehab / ADL 4 - 5  |

| Element Number | RUG3 Group  |
|----------------|---|
| 30             | IA1: Cognitive Impairment / ADL 4 - 5                   |
| 31             | BB2: Behavior Problem with Nursing Rehab / ADL 6 - 10   |
| 32             | BB1: Behavior Problem / ADL 6 – 10                      |
| 33             | BA2: Behavior Problem with Nursing Rehab / ADL 4 - 5    |
| 34             | BA1: Behavior Problem / ADL 4 - 5                       |
| 35             | PE2: Physical Function with Nursing Rehab / ADL 16 - 18 |
| 36             | PE1: Physical Function / ADL 16 – 18                    |
| 37             | PD2: Physical Function with Nursing Rehab / ADL 11 - 15 |
| 38             | PD1: Physical Function / ADL 11 – 15                    |
| 39             | PC2: Physical Function with Nursing Rehab / ADL 9 - 10  |
| 40             | PC1: Physical Function / ADL 9 – 10                     |
| 41             | PB2: Physical Function with Nursing Rehab / ADL 6 - 8   |
| 42             | PB1: Physical Function / ADL 6 - 8                      |
| 43             | PA2: Physical Function with Nursing Rehab / ADL 4 - 5   |
| 44             | PA1: Physical Function / ADL 4 - 5                      |
| 45             | BC1: M3PI group not calculated due to data errors       |
| 46             | RAD: Rehabilitation All Levels / ADL 17 – 18            |
| 47             | RAC: Rehabilitation All Levels / ADL 14 – 16            |
| 48             | RAB: Rehabilitation All Levels / ADL 9 – 13             |
| 49             | RAA: Rehabilitation All Levels / ADL 4 - 8              |

Note that the last 4 entries for RAD, RAC, RAB, and RAA are for the rehabilitation groups in the 34 group model of RUG-III. These values must be present but are not used in the current DLL, which only supports the 44 group model of RUG-III.

## Version History

The version number of each DLL can be determined by examining the file's timestamp (e.g., Version 1.11 DLLs have a file timestamp of 1:11AM). In addition, the DLL version number is returned by the DllVersion parameter described earlier in this documentation.

| Version | R512_16.DLL                         | R512_32.DLL                           |
|---------|-------------------------------------|---------------------------------------|
| 1.00    | Initial release. Supported VB only. | Not publicly released                 |
| 1.10    | Added support for C++.              | Initial release. Supports VB and C++. |



| Version | R512_16.DLL  | R512_32.DLL  |
|---------|--|--|
| 1.11    | If version 1.10 was called from the cDemo16 program, it ran properly only under Windows 95; it failed under Win3.x and NT. This was fixed. | No changes, except that DllVersion returns "1.11". |

## Technical Assistance

If you have questions about the DLL which are not answered by this documentation, you can contact Stepwise Systems by email at [support@stepsys.com](mailto:support@stepsys.com).

## License Agreement

Use of R512\_16.DLL and R512\_32.DLL (hereinafter "SOFTWARE") is contingent on your agreement to the following terms:

### 1. GRANT OF LICENSE

Stepwise Systems, Inc. (hereinafter Stepwise) grants you a limited, non-exclusive license to use the SOFTWARE free of charge.

### 2. DISCLAIMER OF WARRANTY

THE SOFTWARE IS PROVIDED AS IS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, STEPWISE FURTHER DISCLAIMS ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE PRODUCT AND DOCUMENTATION REMAINS WITH RECIPIENT. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL STEPWISE BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, PUNITIVE, OR OTHER DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THIS AGREEMENT OR THE USE OF OR INABILITY TO USE THE PRODUCT, EVEN IF STEPWISE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### 3. SCOPE OF GRANT

You may not reverse engineer, decompile or disassemble the SOFTWARE. Stepwise shall retain title and all ownership rights to the SOFTWARE.

### 4. COPYRIGHT

This SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties.

### 5. MAINTENANCE

Stepwise is not obligated to provide maintenance or updates for the SOFTWARE. However, any maintenance or updates provided by Stepwise shall be covered by this Agreement.

### 6. DISTRIBUTION

The SOFTWARE may be freely distributed provided that it is not modified and the original archive remains intact with all accompanying files, and provided that no fee is charged (except for any reasonable fees necessary to cover

costs of distribution media). Should a fee be charged or any of the distribution package need to be modified, you must contact Stepwise for explicit written permission.

## 7. RUNTIME SOFTWARE

R512\_16.DLL and R512\_32.DLL constitute the RUNTIME SOFTWARE. Stepwise grants you a Limited Runtime License to reproduce and distribute exact copies of the RUNTIME SOFTWARE and to modify, reproduce and distribute the sample application(s) included in this archive if and only if all of the following conditions are satisfied:

- a) You may distribute such copies solely for the purpose of executing specific-purpose applications programs written using an authorized copy of the SOFTWARE;
- b) You remain solely responsible for support, service, upgrades, and technical or other assistance required or requested by anyone receiving such copies;
- c) You may not use the name, logo, or trademark of Stepwise without written permission from Stepwise;
- d) You may not alter, disassemble, decompile, translate, adapt or reverse-engineer the RUNTIME SOFTWARE;
- e) Stepwise makes no warranty other than the warranty provided in this agreement to you; and
- f) You will indemnify Stepwise against any claims or liabilities arising out of its use, reproduction or distribution, and the use by the recipients or copies made hereunder.

## 8. GOVERNING LAW

This agreement is governed by the laws of the State of Texas.