

GURU[®]

GURU[®] 7.1 Release Notes

February 2000

Legal Notices

Copyright © 2004, 2005 Savitar Corporation.

These manuals are provided for the use of customers and customer employees. The entire contents have been copyrighted by Savitar Corporation. Copying, duplicating, selling, or other distribution of the software is a violation of the copyright laws. Reproduction by any means of the software or documentation is prohibited except as permitted in a written agreement with Savitar Corporation.

TITANIUM, GURU, and all other Savitar products including legacy **mdbs** products are copyrighted computer software, and all rights, including intellectual property rights, are reserved by Savitar Corporation. Use of software requires an End User License. Please refer to the End User License Agreement for the corresponding terms and conditions.

TITANIUM is protected by U.S. Patent Nos. 5,611,076 and 5,713,014 in addition to foreign patents and applications pending.

Limitation on Warranties and Liability

The documentation in these manuals is provided for reference purposes only. Savitar Corporation has taken great pains to ensure that the manuals reasonably conform to the software. However, Savitar Corporation reserves the right to make changes in specifications and other information in these publications without prior notice.

Savitar Corporation makes no warranties, either express or implied, respecting the software and documentation or its quality, performance, merchantability, or fitness for any particular purpose. Savitar Corporation software and documentation is licensed for use "as is."

Savitar Corporation shall not be liable to the user or any other person or entity with respect to any liability, loss or damage caused by, or alleged to be caused by, the software and documentation or its use, whether direct, incidental, or consequential, according to the terms of the license agreement.

Trademarks

TITANIUM[®] and GURU[®] are a registered trademarks of Micro Data Base Systems. MDBS IV[®], MDBS III[®], KnowledgeMan[®] and Object/1[®] are also registered trademarks of Micro Data Base Systems.

Other brands and product names may be trademarks or registered trademarks of their respective holders.

Contents

Description	1 : 5
Installation	1 : 5
Program Group Change	1 : 5
ODBC Driver New Features	1 : 6
Indexes on Tables.....	1 : 6
System, User and File DSNs.....	1 : 6
DATABASE Qualifier.....	1 : 6
ActiveX Data Objects	1 : 7
Technical Notes	1 : 8
New CLINK Examples	1 : 8
Java Runtime Version for GRSE	1 : 8
Keymap File.....	1 : 8
TBL Changes	1 : 10
New TBL Sample Program.....	1 : 10
TBL Example for Visual BASIC	1 : 10
New TBL Functions.....	1 : 10
tbl_len().....	1 : 11
tbl_get_y2k2 (), tbl_set_y2k2 ()	1 : 12
tbl_tojul()	1 : 14
tbl_todate().....	1 : 16
Documentation Additions and Errata	1 : 18
Known Functionality Issues	1 : 19
File DSNs	1 : 21
Adding a File DSN (Data Source) for the GURU ODBC Driver.....	1 : 21
Requirements	1 : 21
Steps.....	1 : 21
Using the File DSN from Microsoft Excel	1 : 23

Description

GURU 7.1 is an update to GURU v7.0 and the GURU ODBC Driver v7.0. These release notes are in addition to the GURU 7.0 Release Notes, the GURU ODBC Driver, and the other components of the GURU 7.0 documentation set. Refer to these release notes and the appropriate GURU 7.0 documentation for complete information.

Installation

To ensure proper setup, uninstall GURU 7.0 before installing GURU 7.1. Uninstalling prevents duplicate versions of files when GURU 7.1 is installed in a different directory than GURU 7.0.

Program Group Change

The default location of GURU 7.1 within the Microsoft Windows Start Programs menu has changed. The new location is:

Start → Programs → Micro Data Base Systems, Inc. → GURU 7.

ODBC Driver New Features

Indexes on Tables

The GURU ODBC Driver v7.1 now supports indexes on tables using the built-in GURU functions TOJUL() and LEN(). The ODBC driver uses indexes for rapid data lookups. Previous versions of the GURU ODBC Driver only supported indexes for simple field expressions. Note:

- ◆ Complex expressions using functions other than TOJUL() and LEN() are not supported.
- ◆ The ascending (AZ) or descending (ZA) keyword is required for indexes used with the ODBC driver.

The GURU ODBC Driver automatically recognizes an index using TOJUL() or LEN() when the index is specified in the TBLINFO.DAT file following the name of the table it references. Indexes using TOJUL() or LEN() must be created within GURU; no support for creation of TOJUL() or LEN()-based indexes is provided within the GURU ODBC Driver or the TBL database library.

System, User and File DSNs

The GURU ODBC Driver now supports ODBC System Data Source Names (System DSNs) as well as User DSNs and File DSNs for access to GURU data from ODBC. File DSNs require that the DBQ qualifier be specified for the data source. For more information, see the section “Adding a File DSN (Data Source) for the GURU ODBC Driver” on page 21 of these Release Notes.

DATABASE Qualifier

The GURU ODBC Driver supports the DATABASE qualifier in ODBC connect strings. The DATABASE qualifier can be used to specify the location of the GURU data files in a connect string passed from program control (as when opening an ODBC data source within Visual BASIC). This capability allows access to GURU tables in multiple directories from a single data source. See the ODBC Programmer's Reference for more information on connect strings. When used with GURU ODBC, the DATABASE qualifier has the form:

`DATABASE, path`

The *path* is the fully qualified path name of the directory containing the following files:

- ◆ TBLINFO.DAT
- ◆ The GURU data files required for this invocation of the data source.

ActiveX Data Objects

Enhanced support for ActiveX Data Objects (ADO) provides the ability to display data from GURU BLOB fields on web pages using ADO queries. Note that GURU BLOB data appears within ODBC as long text (SQL_LONGVARCHAR) data.

Technical Notes

New CLINK Examples

Two new CLINK examples show how to call Win32 functions from GURU. These examples are located in the CLINK subdirectory under the GURU installation directory. Since they are compiled DLLs, these examples do not require a C compiler.

- ◆ **SETTITLE** This function sets the title bar in a console mode window. For example, the title bar of an application can be set to the application name rather than to the default “GURU 7”.
- ◆ **NOMOUSE** This example disables GURU mouse functionality for cases where mouse usage is not desired in an application.
- ◆ **MEM32, OFILEDLG, and FTOC** These already-existing examples continue to be available for use in applications. The .IPF file for each example shows how to use the CLINK functions within GURU.

The GURU 7.1 CLINK library continues to require Microsoft Visual C++ v5.0 or above for creation of GURU CLINK functions.

Java Runtime Version for GRSE

The GURU Graphical Rule Set Editor component (GRSE) requires Java Runtime Environment version 1.1 (the version included with GURU 7.1 is JRE 1.1.8).

Note: Java 2 (1.2) is not compatible with GRSE.

Keymap File

GURU 7.1 includes a new keymap file (GURUKEY.DAT) for advanced key remapping under Win32 Console Mode. This file provides an extra key mapping capability in addition to the already-existing key-related commands in GURU 7.0, such as the KEYMAP() function.

Use the already-existing commands for most key-related operations: they are backwards-compatible and easier to use. Use the new facility only for key mappings that cannot be obtained with the already-existing key commands. Limitations of GURUKEY.DAT include:

- ◆ Not backwards-compatible with existing GURU commands.
- ◆ Win32-specific.

- ◆ Not portable to other GURU environments.

The GURUKEY.DAT file is located in the directory named in the GURUHOME environment variable. For more details on its operation, see the comments in the GURUKEY.DAT file.

Printing under Windows 95/98

To print under Windows 95/98, ensure that the current directory (set when starting GURU or using the GURU CD command) has a valid DOS path (directory/filenames with 8 characters or less, no spaces, total length of path less than 255 characters).

This addresses an issue opening Windows 95/98 printers that appears within GURU as a "Printer not ready" error.

TBL Changes

New TBL Sample Program

A new sample program, `SAMTOJUL.C`, is located in the TBL subdirectory. This program shows the use of several new GURU 7.1 TBL functions. To use `SAMTOJUL.C`:

1. Run `MAKSAMPL.BAT` to make all the TBL samples including `MAKETABS.EXE` and `SAMTOJUL.EXE`
2. Run `MAKETABS.EXE` to create TBL data files.
3. Run the GURU program file `SAMTOJUL.IPF` (located in the TBL subdirectory):

```
GURU -G SAMTOJUL
```

This step creates the additional index file used by `SAMTOJUL.EXE`.

4. Run `SAMTOJUL.EXE` to see this sample.

TBL Example for Visual BASIC

An example program showing how to call the TBL C library from Microsoft Visual BASIC is added to the TBL library for C language access to GURU data files. This example is located in the VB subdirectory of the TBL subdirectory under the main GURU installation directory.

Not all TBL functionality is supported from Visual BASIC. Access from Visual BASIC is on a per-field basis (requiring use of functions such as `tbl_getf()`).

New TBL Functions

Several new functions have been added to the TBL library for C language access to GURU data files. These functions are detailed on the following pages.

Name: **tbl_len()**
TBL equivalent of GURU LEN() function.

Synopsis:

```
#include "tbl.h"
double * tbl_len(char * str);
```

Return Values:

Returns a pointer to a `double` representing the length of the passed-in character string.

Arguments:

`str` - Pointer to a null-terminated character string.

Description:

This function returns the length of the passed-in character string `str` consistent with the built-in GURU function `LEN()`. This function is provided for use with indexes using the `LEN()` function within TBL and the GURU ODBC Driver.

Example:

```
#include "tbl.h"
char * s;

/* tbl_len returns static pointer */

s = "abcdef";
printf("Length of [%s] = %lf\n", s, *(tbl_len(s)));
```

Name: `tbl_get_y2k2 ()`, `tbl_set_y2k2 ()`

TBL equivalents for the GURU E.Y2K2 environment variable.

Synopsis:

```
#include "tbl.h"
int tbl_get_y2k2(void);
int tbl_set_y2k2(int thresh);
```

Return Values:

Returns an integer containing the threshold value for two-digit date interpretation in the `tbl_tojul()` function.

Arguments:

`tbl_get_y2k2()` – none.

`tbl_set_y2k2(thresh)` – integer containing the desired threshold value.

Description:

This function sets an internal threshold value within TBL, similar to the E.Y2K2 environment variable within GURU. This threshold value controls the century interpretation of character dates containing a 2-digit year by the TBL function `tbl_tojul()`.

By default the threshold value is 50. 2-digit years less than 50 are interpreted as 21st-century dates—for example, “10/10/49” converts to the Julian date value for October 10, 2049. 2-digit years equal to or greater than the threshold are interpreted as 20th-century dates—for example, “10/10/50” converts to the Julian date value for October 10, 1950.

The function `tbl_set_y2k2()` is used to set the threshold value. For example, setting the threshold value to 38 causes a 2-digit year equal to 38 to be interpreted as 1938, while a 2-digit year of 37 is interpreted as 2037. The function `tbl_get_y2k2()` reads the threshold value. These functions are provided for use with indexes using the `TOJUL()` function within TBL and GURU ODBC Driver.

For more details of the E.Y2K2 environment variable and the `TOJUL()` function, see the GURU documentation.

Example:

```
#include "tbl.h"
int threshold;

threshold = tbl_get_y2k2();
printf("Threshold is %d\n", threshold);

threshold = tbl_set_y2k2(38);
printf("Threshold is now %d\n", threshold);
```

Name: **tbl_tojul()**
TBL equivalent of GURU TOJUL() function.

Synopsis:

```
#include "tbl.h"
double * tbl_tojul(char fmt, char * datestr)
```

Return Values:

Returns a pointer to a `double` representing the Julian date equivalent of the passed-in `datestr` character string. The double contains the value zero (0.0) if an error occurs in the date conversion.

Arguments:

`fmt` – one of the characters 'M', 'D', or 'Y' indicates whether `datestr` begins with the month ('M'), day ('D'), or year ('Y') portion of the date. Use of this format is consistent with the GURU E.DATE environment variable.

`datestr` – null-terminated character string containing a character representation of a calendar date.

Description:

This function returns the Julian date equivalent of the passed-in character string `datestr` consistent with the built-in GURU function TOJUL(). This function is provided for use with indexes using the TOJUL() function within TBL and the GURU ODBC Driver.

The character string has the following forms:

- ◆ MM/DD/YY or MM/DD/YYYY ('M' format)
- ◆ DD/MM/YY or DD/MM/YYYY ('D' format)
- ◆ YY/MM/DD or YYYY/MM/DD ('Y' format)

If the year component is a 2-digit year, see the description of the functions `tbl_get_y2k2()` and `tbl_set_y2k2()` for a discussion of the century interpretation of 2-digit dates.

For more information on the TOJUL() function, see the Functions section of the Programmer's Reference in the GURU documentation.

Example:

```
#include "tbl.h"
/*****
 * ctoj - character date to julian date
 *
 * Arguments:  s  pointer to date character string
 *
 * Returns: double value for the date, or 0.0 if error
 *
 *****/
double ctoj(char * s)
{
    double ddate;

    ddate = *tbl_tojul('M', s);

    return (ddate);
}
```

Name: **tbl_todate()**
TBL equivalent of GURU TODATE() function.

Synopsis:

```
#include "tbl.h"
char * tbl_todate(char fmt, char sep, double jdate, int
yd);
```

Return Values:

Returns a pointer to a null-terminated character string representing the character for the passed-in Julian date `jdate`. The returned pointer is NULL if an error occurs in the date conversion.

Arguments:

`fmt`—one of the characters 'M', 'D', or 'Y' indicating whether the returned date begins with the month ('M'), day ('D'), or year ('Y') portion of the date. Use of this format is consistent with the GURU E.DATE environment variable.

`sep`—character for separating month, year, and day portions of the returned date.

`jdate`—double containing the Julian date to convert.

`yd` - integer indicating the number of digits to use in the year portion of the returned date.

Description:

This function returns a character string representing a date, consistent with the built-in GURU function TODATE(). This function is provided for use with indexes using the TOJUL() function within TBL and the GURU ODBC Driver.

The character string has the following forms:

- ◆ MM/DD/YY or MM/DD/YYYY ('M' format)
- ◆ DD/MM/YY or DD/MM/YYYY ('D' format)
- ◆ YY/MM/DD or YYYY/MM/DD ('Y' format)

Although the separator character can be any character, the '/' character is normally used.

The Julian date is a numeric representation of a date consisting of the number of days since the base date (October 15, 1582 A.D.). The maximum date is December 31, 9999 A.D. This date is consistent with the value returned by the GURU TOJUL() function and the TBL `tbl_tojul()` function.

For more information about the TODATE() and TOJUL() functions, see the Functions section of the Programmer's Reference in the GURU documentation.

Example:

```
#include "tbl.h"
/*****
 * jtoc - julian date to character date
 *
 * Returns:    pointer to date string in a static location
 *
 *****/
char *jtoc(double ddate)
{
    return tbl_todate('M', '/', ddate, 4);
}
```

Documentation Additions and Errata

The following items have been noted during the testing of **GURU 7.1** as errata or necessary additions in the GURU 7.0 documentation (PDRs):

- LAN Installation** Installing LAN GURU in the same directory as a previously existing installation requires the resetting of read-only attributes for executable and other shared files before running INSTALL.
- LAN Uninstall** GURU Uninstall must be run on all workstations prior to running uninstall on the server. A file used in the GURU workstation uninstall process is stored in the GURU directory on the server.
- BMEMADR()** The BMEMADR() function accepts a BLOB field as a parameter, not a file.
- BPUTC()** The correct order for the arguments to the BPUTC() function is:
BPUTC(*blobfield*, *offset*, *nexp*).
- DELSTR()** The DELSTR() function returns the argument string instead of an integer code.
- FCOPY()** Using FCOPY("file", "a:") without a target file name creates a target file name based on the FCOPY temporary file name (for example, FCPY215).
- MATCH()** The MATCH() function returns zero (0) when the second argument is a null string.
- tbl_index()** When passing the array of key structure descriptors as the third parameter to the TBL tbl_index() function, use an extra empty key structure descriptor as a terminator. The name field of the empty key structure descriptor must contain an empty string (" ").

Known Functionality Issues

A number of miscellaneous functionality issues (PSRs) are addressed in GURU 7.1. The following major items are addressed:

Installation	A number of installation and uninstallation issues dealing with setting and removal of the PATH, GURUHOME, and other environment variables are addressed.
LAN Installation	A case of extra files installation on LAN workstations is addressed.
Expert System	A case where the expert system SHOW DIAGRAM and BUILD commands hang is addressed. In addition, some cases are addressed within the Graphical Rule Set Editor (GRSE) where changes are not saved completely or where incomplete rule sets cause a hang.
GURU Debugger	A case is addressed where pressing CTRL-Y (FWHY) during a debugging session with the GURU debugger causes a general protection fault.
Read-only Tables	Support for files marked as read-only by the operating system (ATTRIB +R) is added for table (.ITB), index (.IND), BLOB (.BLB), password (.IGU), and UDF function (.KGB) files.
Table Dates	<p>A case is addressed where the Modified and Created dates displayed by the SHOW <i>tablename</i> command are incorrect for the year 2000.</p> <p>These dates are displayed and stored as 2-digit dates for backward compatibility. These dates are interpreted correctly by TOJUL() and other GURU date functions with the appropriate setting of E.Y2K2.</p> <p>This correction also applies to the TBLINFO function (options 12 and 13) and the tbl_show() function.</p>
Keyboard	A number of issues with backwards-compatible behavior of the keyboard and keypad, especially under Windows 95, is addressed. The value returned by function keys and certain keys on the numeric keypad when the CAPS LOCK and/or NUM LOCK is enabled is not consistent in GURU 7.0, and this issue has been addressed.
TBLFLUSH()	A case is addressed where the TBLFLUSH() function called with no arguments produces a system error.
TODAY()	A case is addressed where the TODAY() function displays the date incorrectly when E.DTYP is set to "Y".

Known Functionality Issues

BLOB data with Microsoft Word	A case where BLOB data used with Microsoft Word (obtained through Microsoft Query) results in a system error is addressed.
Default printer	A case is addressed where the default printer on Windows NT is invalid (#PRNOUT= " ").
Local Printing	<p>A case is addressed where GURU prints slowly to a local printer. GURU 7.1 now buffers data sent to a local printer.</p> <p>Print buffering for local printers can be disabled if the environment variable GURUNOPBUF is set (for example, SET GURUNOPBUF=X).</p>
BTOA()	A case is addressed where many repeated calls to BTOA() causes a system error under Windows 95/98.
ROUND()	A case is addressed where the ROUND() function returns incorrect results for certain input values.
Mouse movement	A case is addressed where a mouse movement causes a TEXTW() text box created with the popup special effect to exit prematurely. Also addressed is a case where mouse movement causes the next page of output from a SELECT statement to be displayed when E.PAUS=TRUE.
ODBC Password Searching	The GURU ODBC Driver now searches for GPASS.IGU in the GURU-HOME directory if GPASS . IGU is not located in the directory where the data source files are located.
ODBC Access Denied Error	A case is addressed within the GURU ODBC Driver where the "Access Denied" error is not reported when the user does not have access to the requested data.
ODBC Multi-field Indexes	A case is addressed within the GURU ODBC Driver query optimization where indexes defined with multiple fields could cause incorrect query results.
ODBC Driver European Language Support	A number of issues are addressed involving support within the GURU ODBC Driver for European languages using 8-bit characters (ASCII characters greater than 127). These include conversion of 8-bit characters within BLOB data and support for usernames and passwords containing 8-bit characters.

File DSNs

Adding a File DSN (Data Source) for the GURU ODBC Driver

This section shows an example for setting up a File DSN for use with the GURU ODBC Driver. This example works for one particular system under Windows 95 using Microsoft Office 97. While other versions of these tools may have slightly different menus or steps, the process is similar.

To use ODBC from Microsoft Query, a File DSN is required. Microsoft Query is the data access component for Word or Excel (Microsoft Office 97 or newer).

The data source information for a File DSN is stored in a file. User DSNs and System DSNs, the other type of data source supported by the GURU ODBC Driver, the data source information is stored in the system registry.

Requirements

Before starting, install the following components on the system:

- ◆ GURU ODBC Driver.
- ◆ Microsoft Office *including* Microsoft Query (Query is an optional component that may not be installed by default).

Steps

Add File DSN

1. Use the instructions in the GURU ODBC Driver document to create a User DSN data source. Stop at Figure 6 on page 8.

Figure 6 shows the ODBC Data Source Administrator main dialog. At this point, the ODBC Data Source has been created.

2. Click the **File DSN** tab of the dialog to create a File DSN for the existing Payroll data source.

By default this dialog creates DSN files in the directory `\Program Files\Common Files\ODBC\Data Sources`.

3. Click the **Look In:** list box and navigate to the `\My Documents` directory.

Since Word and Excel search for DSN files in the document's default

directory, this example creates the DSN file in the directory \My Documents.

4. Click the **Add** button. The Create New Data Source dialog appears.

Specify the DBQ qualifier

1. Select the GURU ODBC Driver line and click the **Advanced** button. The Advanced File DSN Creation dialog appears.

2. Click once in the text edit window.

3. Add a line containing the DBQ qualifier.

This line contains the full path name for the data source previously defined. For example, if the path name is C:\DATA add the line:

```
DBQ=C:\DATA
```

4. Click **OK** to finish the Advanced dialog.

Name the File DSN

1. Click **Next** to name the DSN file.
2. Type the name, such as "Payroll", in the entry field.
3. Click **Next**. The final dialog appears to allow verification of the given information.
4. Click on **Finish** to save this File DSN with the information as specified in previous steps.
5. If the **Verify Connection** check box was checked (default) in step 6, the GURU ODBC Logon dialog appears. Either:
 - ◆ Enter the user/password.
 - ◆ Click **OK** if there is no password for this data source.

The completed File DSN now appears in the My Documents directory.

6. Click **OK** to exit the ODBC Administrator.

Thses steps complete the setup of the File DSN.

Using the File DSN from Microsoft Excel

Once the File DSN is set up, it can be used from Microsoft Excel 97, assuming Microsoft Query is installed:

1. In the main window of the Microsoft Excel spreadsheet, click **Data** → **Get External Data** → **Create New Query** from the main menu bar.

The Choose Data Source dialog appears.

2. With <New Data Source> selected in the Databases list, click **Browse**.

The Browse Data Sources dialog appears, showing the files in the My Documents directory.

3. Select the `Payroll.dsn` file and click **Open**.

“Payroll” now appears as the selected choice in the Choose Data Sources dialog.

4. Click **OK** to open the `Payroll.dsn` file.

5. When the GURU ODBC Driver Logon dialog appears, do one of the following:

- ◆ Enter the user/password.
- ◆ Click **OK** if there is no password to enter.

The Query Wizard dialog appears showing tables from the data source.

6. Choose the tables and columns as prompted.

7. Click Finish in the Query Wizard dialog.

The data is returned to Microsoft Excel.