

G2-ODBC Bridge

Release Notes
Version 12.0 Rev. 1



G2-ODBC Bridge Release Notes, Version 12.0 Rev. 1

May 2020

The information in this publication is subject to change without notice and does not represent a commitment by Gensym Corporation.

Although this software has been extensively tested, Gensym cannot guarantee error-free performance in all applications. Accordingly, use of the software is at the customer's sole risk.

Copyright © 1985-2020 Gensym Corporation

All rights reserved. No part of this document may be reproduced, stored in a retrieval system, translated, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Gensym Corporation.

Gensym®, G2®, Optegrity®, and ReThink® are registered trademarks of Gensym Corporation.

NeurOn-Line™, Dynamic Scheduling™, G2 Real-Time Expert System™, G2 ActiveXLink™, G2 BeanBuilder™, G2 CORBALink™, G2 Diagnostic Assistant™, G2 Gateway™, G2 GUIDE™, G2GL™, G2 JavaLink™, G2 ProTools™, GDA™, GFI™, GSI™, ICP™, Integrity™, and SymCure™ are trademarks of Gensym Corporation.

Telewindows is a trademark or registered trademark of Microsoft Corporation in the United States and/or other countries. Telewindows is used by Gensym Corporation under license from owner.

This software is based in part on the work of the Independent JPEG Group.

Copyright © 1998-2002 Daniel Veillard. All Rights Reserved.

SCOR® is a registered trademark of PRTM.

License for Scintilla and SciTE, Copyright 1998-2003 by Neil Hodgson, All Rights Reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

All other products or services mentioned in this document are identified by the trademarks or service marks of their respective companies or organizations, and Gensym Corporation disclaims any responsibility for specifying which marks are owned by which companies or organizations.

Ignite Technologies, Inc.
401 Congress Ave., Suite 2650
Austin, TX 78701 USA
Telephone: +1-800-248-0027
Email: success@ignitetech.com

Part Number: DOC048-1200

Contents

| | | |
|------------------|---------------------------------------|----------|
| | Preface | v |
| | About these Notes | v |
| | Audience | v |
| | Conventions | vi |
| | Related Documentation | vii |
| | Customer Support Services | x |
| Chapter 1 | Overview of the G2-ODBC Bridge | 1 |
| | Introduction | 1 |
| | Obtaining Bug Fix Information | 1 |
| Chapter 2 | Requirements | 3 |
| | Introduction | 3 |
| | System Requirements | 4 |
| | Third-Party Software Requirements | 4 |
| | 32-Bit ODBC Drivers | 4 |
| | 64-Bit ODBC Drivers | 4 |
| | ODBC Driver Conformance Levels | 5 |
| | Release Compatibility | 5 |
| | G2 | 6 |
| | ODBC | 6 |
| Chapter 3 | Release Features | 7 |
| | Introduction | 7 |
| | G2-ODBC Bridge Version 12.0 Rev. 0 | 8 |
| | G2-ODBC Bridge Version 11.0 Rev. 0 | 8 |
| | G2-ODBC Bridge Version 8.3 Rev. 0 | 8 |
| | G2-ODBC Bridge Version 8.2 Rev. 0 | 8 |
| | G2-ODBC Bridge Version 7.1 Rev. 1 | 9 |

| | | |
|------------------|---|-----------|
| Chapter 4 | G2-ODBC Bridge-Specific Information | 11 |
| | Introduction | 11 |
| | ODBC Overview | 11 |
| | Structured Query Language | 12 |
| | Connecting to the Database | 13 |
| | DNS Configuration via an RPC | 17 |
| | Interface Object Settings | 19 |
| | Auto-Commit | 19 |
| | User Names and Passwords | 19 |
| | Accessing Database Data | 20 |
| | Database Character Sets and Text Conversion | 20 |
| | Database Queries | 20 |
| | Cursor Objects and Bind Variables | 20 |
| | Stored Procedures Returning Record Sets | 21 |
| | Event Notification through Triggers | 22 |
| | Trigger Installation | 23 |
| | Trigger Deinstallation | 23 |
| | Column Attribute Naming and Aliases | 24 |
| | Using the G2-ODBC Bridge with Stored Procedures | 24 |
| | Example: Using db-exec-stored-proc | 26 |
| | Example: Using db-exec-stored-proc-return | 28 |
| | Index | 29 |

Preface

Describes these release notes and the conventions that they use.

| | |
|---------------------------|-----|
| About these Notes | v |
| Audience | v |
| Conventions | vi |
| Related Documentation | vii |
| Customer Support Services | x |



About these Notes

These Release Notes:

- Explain how to install your G2-ODBC Bridge, and how to establish connections to databases and communicate with them. This information is unique to ODBC and is not included in the *G2 Database Bridge User's Guide*.
- Describe the system requirements for this release.
- Give information on all new and changed bridge features.

Audience

To use this manual, you must have at least a limited knowledge of G2, and a thorough understanding of the database system that you want to connect to G2.

Conventions

This guide uses the following typographic conventions and conventions for defining system procedures.

Typographic

| Convention Examples | Description |
|--|--|
| g2-window, g2-window-1, ws-top-level, sys-mod | User-defined and system-defined G2 class names, instance names, workspace names, and module names |
| history-keeping-spec, temperature | User-defined and system-defined G2 attribute names |
| true, 1.234, ok, "Burlington, MA" | G2 attribute values and values specified or viewed through dialogs |
| Main Menu > Start KB Workspace > New Object create subworkspace Start Procedure | G2 menu choices and button labels |
| conclude that the x of y ... | Text of G2 procedures, methods, functions, formulas, and expressions |
| <i>new-argument</i> | User-specified values in syntax descriptions |
| <u>text-string</u> | Return values of G2 procedures and methods in syntax descriptions |
| File Name, OK, Apply, Cancel, General, Edit Scroll Area | GUIDE and native dialog fields, button labels, tabs, and titles |
| File > Save Properties | GMS and native menu choices |
| workspace | Glossary terms |

| Convention Examples | Description |
|--|-------------------------------------|
| c:\Program Files\Gensym\ /usr/gensym/g2/kbs | Windows pathnames UNIX pathnames |
| spreadsh.kb | File names |
| g2 -kb top.kb | Operating system commands |
| public void main() gsi_start | Java, C and all other external code |

Note Syntax conventions are fully described in the *G2 Reference Manual*.

Procedure Signatures

A procedure signature is a complete syntactic summary of a procedure or method. A procedure signature shows values supplied by the user in *italics*, and the value (if any) returned by the procedure *underlined*. Each value is followed by its type:

```
g2-clone-and-transfer-objects
(list: class item-list, to-workspace: class kb-workspace,
 delta-x: integer, delta-y: integer)
-> transferred-items: g2-list
```

Related Documentation

G2 Core Technology

- *G2 Bundle Release Notes*
- *Getting Started with G2 Tutorials*
- *G2 Reference Manual*
- *G2 Language Reference Card*
- *G2 Developer's Guide*
- *G2 System Procedures Reference Manual*

- *G2 System Procedures Reference Card*
- *G2 Class Reference Manual*
- *Telewindows User's Guide*
- *G2 Gateway Bridge Developer's Guide*

G2 Utilities

- *G2 ProTools User's Guide*
- *G2 Foundation Resources User's Guide*
- *G2 Menu System User's Guide*
- *G2 XL Spreadsheet User's Guide*
- *G2 Dynamic Displays User's Guide*
- *G2 Developer's Interface User's Guide*
- *G2 OnLine Documentation Developer's Guide*
- *G2 OnLine Documentation User's Guide*
- *G2 GUIDE User's Guide*
- *G2 GUIDE/UII Procedures Reference Manual*

G2 Developers' Utilities

- *Business Process Management System Users' Guide*
- *Business Rules Management System User's Guide*
- *G2 Reporting Engine User's Guide*
- *G2 Web User's Guide*
- *G2 Event and Data Processing User's Guide*
- *G2 Run-Time Library User's Guide*
- *G2 Event Manager User's Guide*
- *G2 Dialog Utility User's Guide*
- *G2 Data Source Manager User's Guide*
- *G2 Data Point Manager User's Guide*
- *G2 Engineering Unit Conversion User's Guide*
- *G2 Error Handling Foundation User's Guide*
- *G2 Relation Browser User's Guide*

Bridges and External Systems

- *G2 ActiveXLink User's Guide*
- *G2 CORBALink User's Guide*
- *G2 Database Bridge User's Guide*
- *G2-ODBC Bridge Release Notes*
- *G2-Oracle Bridge Release Notes*
- *G2-Sybase Bridge Release Notes*
- *G2 JMail Bridge User's Guide*
- *G2 Java Socket Manager User's Guide*
- *G2 JMSLink User's Guide*
- *G2 OPCLink User's Guide*
- *G2 PI Bridge User's Guide*
- *G2-SNMP Bridge User's Guide*
- *G2 CORBALink User's Guide*
- *G2 WebLink User's Guide*

G2 JavaLink

- *G2 JavaLink User's Guide*
- *G2 DownloadInterfaces User's Guide*
- *G2 Bean Builder User's Guide*

G2 Diagnostic Assistant

- *GDA User's Guide*
- *GDA Reference Manual*
- *GDA API Reference*

Customer Support Services

You can obtain help with this or any Gensym product from Gensym Customer Support. Help is available online, by telephone and by email.

To obtain customer support online:

➔ Access Ignite Support Portal at <https://support.ignitetechnology.com>.

You will be asked to log in to an existing account or create a new account if necessary. Ignite Support Portal allows you to:

- Register your question with Customer Support by creating an Issue.
- Query, link to, and review existing issues.
- Share issues with other users in your group.
- Query for Bugs, Suggestions, and Resolutions.

To obtain customer support by telephone or email:

➔ Use the following numbers and addresses:

United States Toll-Free +1-855-453-8174

United States Toll +1-512-861-2859

Email support@ignitetechnology.com

Overview of the G2-ODBC Bridge

Describes the G2-ODBC Bridge.

Introduction 1

Obtaining Bug Fix Information 1



Introduction

The G2-ODBC Bridge provides connectivity between G2 and databases by using Microsoft's ODBC interface. The G2-ODBC Bridge works in conjunction with other software supplied by Gensym and third-party vendors. Although the G2-ODBC Bridge may provide connectivity to any third-party product that has an ODBC driver, the bridge has only been tested with interfaces to relational databases.

The G2-ODBC Bridge is based on the G2-Database Bridge common architecture. This document supplements the *G2 Database Bridge User's Guide*.

The G2-ODBC Bridge:

- Provides additional features and enhancements to the G2-Oracle Bridge.
- Fixes various bugs in earlier versions of the bridge.

Obtaining Bug Fix Information

This release includes various bug fixes for G2-ODBC Bridge. These fixes are documented on Ignite Support Portal at <https://support.ignitetech.com>.

Requirements

Describes the system and third-party requirements of this bridge.

| | |
|-----------------------------------|---|
| Introduction | 3 |
| System Requirements | 4 |
| Third-Party Software Requirements | 4 |
| Release Compatibility | 5 |



Introduction

You install the G2-ODBC Bridge as part of the G2 Bundle.

The following version-specific materials comprise the G2-ODBC Bridge:

| Product | Version |
|--|---------------------|
| <i>G2 Database Bridge User's Guide</i> | Version 12.0 Rev. 0 |
| <i>G2-ODBC Bridge Release Notes</i> | Version 12.0 Rev. 0 |
| <code>g2-odbc.exe</code> | Version 12.0 Rev. 0 |
| <code>g2-database.kb</code> | Version 12.0 Rev. 0 |

Note To achieve a successful installation, you must use only materials in this release. You must not use materials from previous releases in conjunction with this release or unexpected results may occur.

System Requirements

The system requirements for the G2-ODBC Bridge are Windows 7/8.1/10 / Server 2008 / Server 2008 R2 / Server 2010 with 2.5 MBytes of disk space, and a 32-bit/64-bit (Enterprise only) ODBC driver to connect to your database.

Third-Party Software Requirements

The G2-ODBC Bridge has specific compatibility requirements with ODBC.

32-Bit ODBC Drivers

Drivers are the software interface components of ODBC which process requests from the G2-ODBC Bridge and send and return data from a database. The ODBC Driver Manager is used to add and remove drivers from your system.

Note ODBC drivers and the ODBC manager are not included with the G2-ODBC Bridge. You must contact your database reseller for information on obtaining ODBC drivers for your database.

The G2-ODBC Bridge shipped with G2 Standard Edition *only support 32-bit ODBC drivers*. In order to load and configure 32-bit ODBC drivers, you will also need a 32-bit ODBC 3 Driver Manager. Typically the driver manager is included with the ODBC driver packs.

Note The 32-bit ODBC manager for 64-bit Windows systems is located at C:\Windows\SysWOW64\odbcad32.exe. It's not accessible through the Control Panel. If you're using G2 Standard Edition on 64-bit Windows systems, please use the 32-bit ODBC manager to configure your data sources based on 32-bit ODBC drivers.

64-Bit ODBC Drivers

The G2-ODBC Bridge shipped with G2 Version 2015 Enterprise *support both 32-bit and 64-bit ODBC drivers*. The ODBC manager accessible through Control Panel can be used to configure 64-bit ODBC data sources based on 64-bit ODBC drivers.

ODBC Driver Conformance Levels

The ODBC specification defines the supported functionality for ODBC drivers. The specification also categorizes this functionality into various degrees of conformance to the ODBC standard for both ODBC API and ODBC SQL grammar. This categorization of functionality is collectively referred to as conformance levels.

To claim that a driver conforms to a particular level, it must support all of the functionality of that level. Conformance with a given level implies complete conformance with all lower levels. The ODBC API defines a set of core functions that correspond to the functions in the X/Open and SAG CLI specifications. ODBC also defines two extended sets of functionality, referred to as Level 1 and Level 2 conformance.

The latest version of ODBC available is ODBC 3. The ODBC conformance levels of ODBC 3 have different requirements than the ODBC API conformance levels of ODBC 2. In particular, all the features implied by ODBC 2 API conformance level 1 are now part of the core conformance level of the ODBC 3 API.

A Version 2 ODBC driver:

- Must meet level 1 conformance to use all bridge functionality, with the exception of bind variables.
- Must meet level 2 conformance to use bind variables.

A Version 3 ODBC driver:

- Must meet core level conformance to use all bridge functionality, with the exception of bind variables.
- Must meet level 2 conformance in order to utilize bind variables.

If you attempt to utilize bind variables with an ODBC driver version that does not conform to the appropriate level, you will receive a “driver does not support this function” error.

Note The G2-ODBC Bridge only supports ODBC Driver Manager Version 3 or higher.

Release Compatibility

The G2-ODBC Bridge has specific compatibility requirements with G2 and ODBC.

It is not necessary to have a separately installed G2 Gateway to run this version of the bridge.

G2

The G2 bridges ship with a bridge KB named `g2-database.kb`, which will load with G2 Version 6.x or later on any platform where G2 is supported.

To load the bridge KBs into G2, it must be authorized with the Online option or similar G2 Gateway and ICP options.

ODBC

In G2 Standard, the bridge executable is a 32-bit application and is only compatible with the 32-bit ODBC driver manager and 32-bit ODBC drivers. To obtain 32-bit ODBC drivers for your database, please consult your database vendor.

A knowledge base built to use previous versions of the G2-ODBC Bridge *will not be compatible* with G2-ODBC Bridge Version 12.0 Rev. 0. You must convert your knowledge base application as specified in these release notes and in the *G2 Database Bridge User's Guide*.

Release Features

Describes the new and changed features in this release.

| | |
|------------------------------------|---|
| Introduction | 7 |
| G2-ODBC Bridge Version 12.0 Rev. 0 | 8 |
| G2-ODBC Bridge Version 11.0 Rev. 0 | 8 |
| G2-ODBC Bridge Version 8.3 Rev. 0 | 8 |
| G2-ODBC Bridge Version 8.2 Rev. 0 | 8 |
| G2-ODBC Bridge Version 7.1 Rev. 1 | 9 |
| G2-ODBC Bridge Version 7.1 Rev. 0 | 9 |



Introduction

This chapter describes the new features and changes in the G2-ODBC Bridge for the last several releases of the bridge.

The following table shows the G2-ODBC Bridge versions that these release notes cover and their corresponding G2 Bundle versions:

| G2-ODBC | G2 Bundle Version |
|---------------------|--------------------------|
| Version 12.0 Rev. 0 | Version 2015 |
| Version 11.0 Rev. 0 | Version 2011 |
| Version 8.3 Rev. 0 | Version 8.3 Rev. 0 |

| G2-ODBC | G2 Bundle Version |
|--------------------|--------------------------|
| Version 8.2 Rev. 0 | Version 8.2 Rev. 0 |
| Version 7.1 Rev. 1 | Version 8.1 Rev. 1 |
| Version 7.1 Rev. 0 | Version 8.1 Rev. 0 |

G2-ODBC Bridge Version 12.0 Rev. 0

Currently G2-ODBC Bridge Version 12.0 Rev. 0 has no essential change beyond Version 11.0 Rev. 0.

G2-ODBC Bridge Version 11.0 Rev. 0

The changes made to the G2-ODBC Bridge Version 11.0 Rev. 0 involved modifications to both `g2-odbc.exe` and `g2-database.kb`. As a result, you must upgrade `g2-database.kb` to use the G2-ODBC Bridge Version 11.0 Rev. 0.

The G2-ODBC UNICODE Bridge is now provided. It's build with UNICODE support, so it's possible to handle Unicode strings in Database queries.

Both 32-bit and 64-bit G2-ODBC Bridges were provided now, therefore customer can use 64-bit ODBC drivers when necessary.

G2-ODBC Bridge Version 8.3 Rev. 0

The changes made to the G2-ODBC Bridge Version 8.3 Rev. 0 involved modifications to both `g2-odbc.exe` and `g2-database.kb`. As a result, you must upgrade `g2-database.kb` to use the G2-ODBC Bridge Version 8.3 Rev. 0.

The G2-ODBC Bridge now requires `sys-mod.kb` and `uilroot.kb`.

The G2-ODBC Bridge supports DNS configuration through an RPC. For details, see [DNS Configuration via an RPC](#).

The G2-ODBC Bridge supports stored procedures returning record sets. For details, see [Stored Procedures Returning Record Sets](#).

The G2-Oracle Bridge supports "smart fetch." For details, see the *G2 Database Bridge User's Guide*.

G2-ODBC Bridge Version 8.2 Rev. 0

G2-ODBC Bridge has been upgraded to Version 8.2 Rev. 0 for the G2 Bundle Version 8.2 Rev. 0. The bridge has been compiled with G2 Gateway Version 8.2.

The G2-ODBC Bridge has one new feature, which is available for all G2 Database Bridges:

- Three new attributes on the `g2-database-interface` class for inserting NULLs into tables:
 - `set-null-string`
 - `set-null-number`
 - `set-null-options`

For details, see the *G2 Database Bridge User's Guide*.

G2-ODBC Bridge Version 7.1 Rev. 1

The G2-ODBC Bridge Version 7.1 Rev. 1 provides the `-shutdown_on_disconnect` command-line option.

For details, see the *G2 Database Bridge User's Guide*.

G2-ODBC Bridge Version 7.1 Rev. 0

On Windows platforms, you can determine the bridge version without having to run G2 by choosing Properties on the bridge executable file, for example, `g2-odbc.exe`, and clicking the Version tab. The Product Version attribute indicates the bridge version.

G2-ODBC Bridge- Specific Information

Provides information specific to the G2-ODBC database.

| | |
|---|----|
| Introduction | 11 |
| ODBC Overview | 11 |
| Structured Query Language | 12 |
| Connecting to the Database | 13 |
| Accessing Database Data | 20 |
| Event Notification through Triggers | 22 |
| Column Attribute Naming and Aliases | 24 |
| Using the G2-ODBC Bridge with Stored Procedures | 24 |



Introduction

This chapter contains G2-ODBC Bridge documentation unique to ODBC and not included in the *G2 Database Bridge User's Guide*. This information is not necessarily related to the release of a new version of the G2-ODBC Bridge.

ODBC Overview

The G2-ODBC Bridge is designed to communicate with any database and on any platform that provides an ODBC driver. The G2-ODBC Bridge works in conjunction with other software components in order to provide communication with databases.

The G2-ODBC Bridge directly communicates with the ODBC driver manager. The driver manager contains ODBC database driver configuration information, which is identified by a Data Source Name (DSN). The DSN is a unique name, which you specify in the `database-connect-string` attribute of the `g2-database-interface` object. Typically, connection information specific to the database communication transport (SQL*Net, EDA Link, I-Net, etc.) such as username, password, and connection string are stored in the configuration identified by the DSN.

Drivers are the software interface components of ODBC which process requests from the bridge and return data. The ODBC driver manager is used to add and delete drivers from your system.

Structured Query Language

Structured Query Language (SQL) is the language that is used with ODBC to communicate with a database. ODBC conforms to the 1992 X/Open and SQL Access Group (SAG) SQL CAE specification (SQL/92 ANSI Standard). Although many database vendors claim conformance to the SQL/92 standard, most only conform to a certain degree. This means that there may be some differences in SQL syntax among database vendors. Since ODBC is 100% conformant to the SQL/92 syntax standard, you must provide equivalent syntax when performing any database transaction.

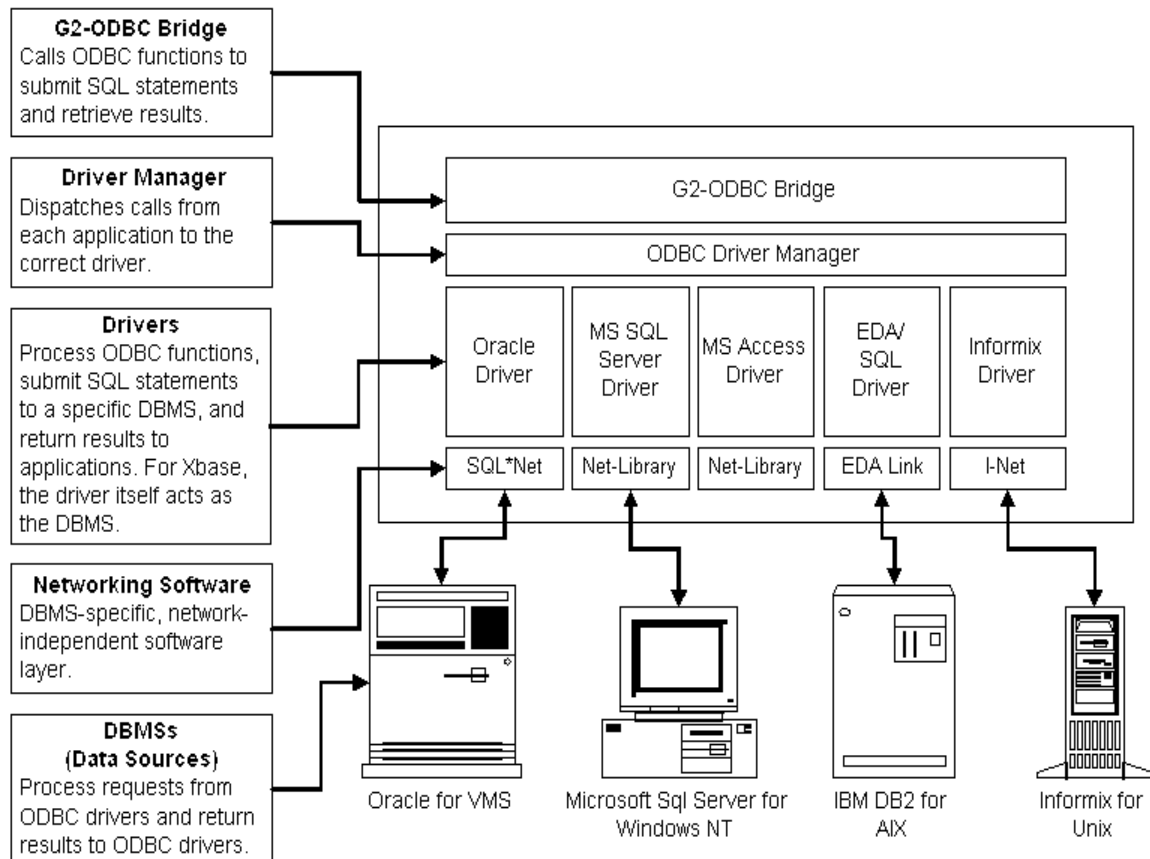
Please consult the SQL/92 ANSI standard available at:

<http://web.ansi.org>

The address of the American National Standards Institute is:

11 West 42nd Street
New York, NY 10036

Connecting to the Database



The figure above shows the relationship between the G2-ODBC Bridge and a database. The G2-ODBC Bridge communicates directly with the ODBC driver manager. The ODBC driver manager dynamically loads database data source drivers and performs ODBC initialization. You must create definitions within the ODBC driver manager for each data source driver you want to connect to.

Note You must install the ODBC driver for a database before you can define the data source.

The data source driver performs the actual communication with the database. A data source includes the data that you wish to access (the database) and the information needed to get to that data.

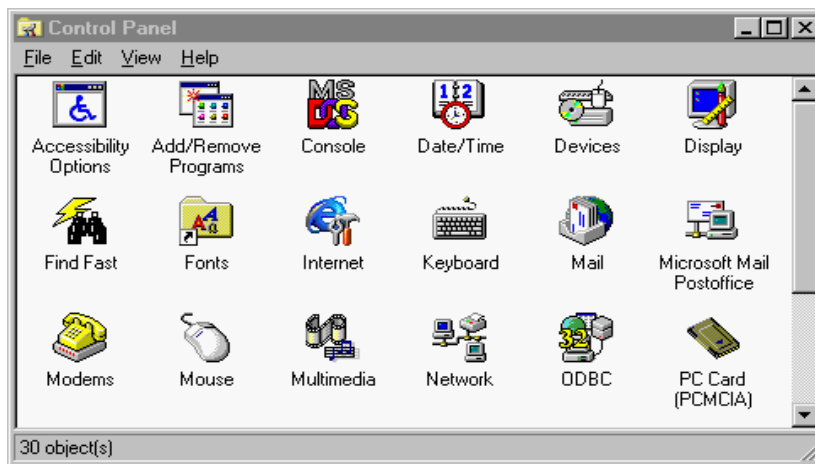
Examples of data sources:

- An MS SQL Server database, the server on which it resides, and the network used to access the server.
- A directory containing a set of dBASE files that you want to access.

The ODBC driver manager is automatically installed if you choose to install ODBC support during the installation of your database or during the installation of any other product that supports ODBC. However, *you must utilize a 32-bit ODBC driver manager*.

Once database installation is complete, you must configure an ODBC data source driver alias with the ODBC driver manager. An ODBC driver alias is a name, provided by you, which refers to a specific ODBC driver configuration. This configuration typically contains database connection information such as user name and password. The ODBC driver alias must be entered into the `database-connect-string` attribute of the `g2-database-interface` object.

To run the ODBC driver manager, double-click on the ODBC icon located under the Control Panel in the `Main` program group. If you have manually installed the 32-bit driver manager, the icon may appear in its own program group. The ODBC driver manager icon will contain 32 indicating that it is a 32-bit manager.



Data source driver aliases must be configured to provide connectivity to databases. When you start the 32-bit ODBC driver manager, you will be presented with a list of 32-bit data source driver aliases currently configured on your system. If you have not configured any ODBC driver aliases, the Data Sources dialog will be empty.

Note If you start a 16-bit ODBC driver manager, the 32-bit ODBC drivers will not be displayed.

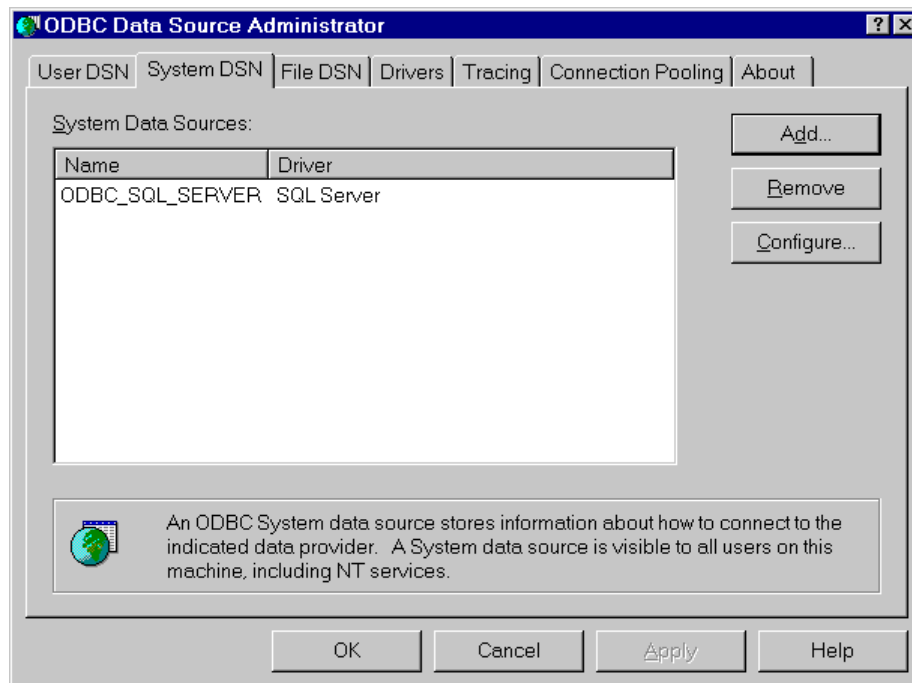
Windows provides two methods for installing ODBC drivers:

- User Data Sources (User DSN)
- System Data Sources (System DSN)

User DSN represents a library of data sources available to the current user. System DSN represents a library of data sources available to the local machine. By default, the User DSN dialog is presented when the ODBC driver manager is accessed. To enter the System DSN configuration, the System DSN tab, located at the top of the dialog, must be selected. If the bridge will be run from an account other than the account configuring ODBC drivers, or if the bridge will be started by a system utility or as a Windows service, the ODBC drivers must be configured as System DSNs. If this is not done, an error similar to the following may occur:

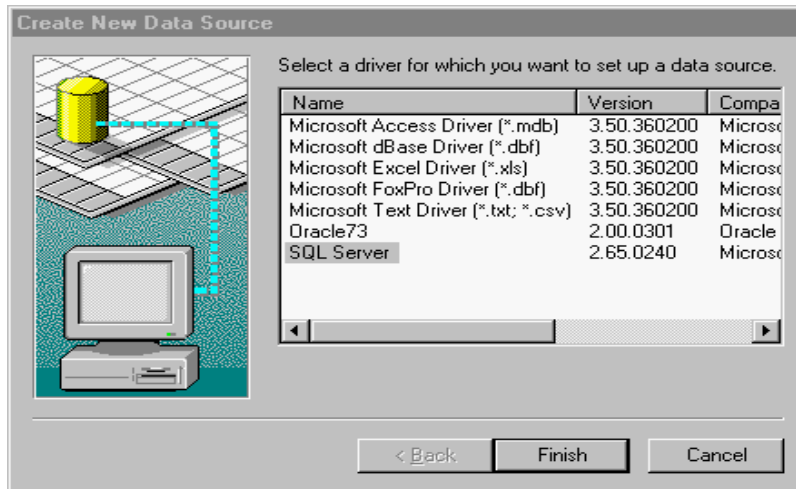
```
G2-ODBC: <context-name>: ERROR: Could not connect to database.
```

```
G2-ODBC: <context-name>: ERROR: ss=IM002 [Microsoft]
[ODBC Driver Manager] Data source name not found and
no default driver specified
```

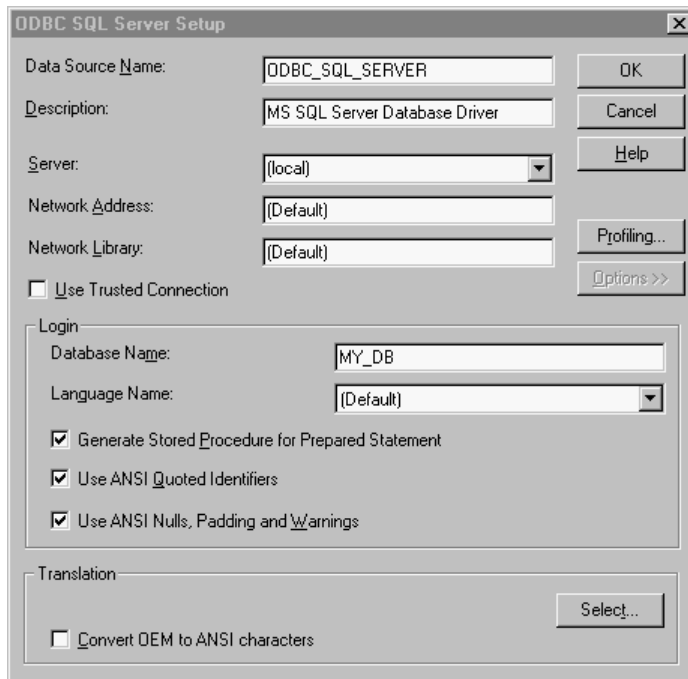


To configure a data source driver alias, click on the Add button in the Data Sources dialog. You will then be presented with a list of data source drivers currently installed on your system. If you need a driver that is not on the list,

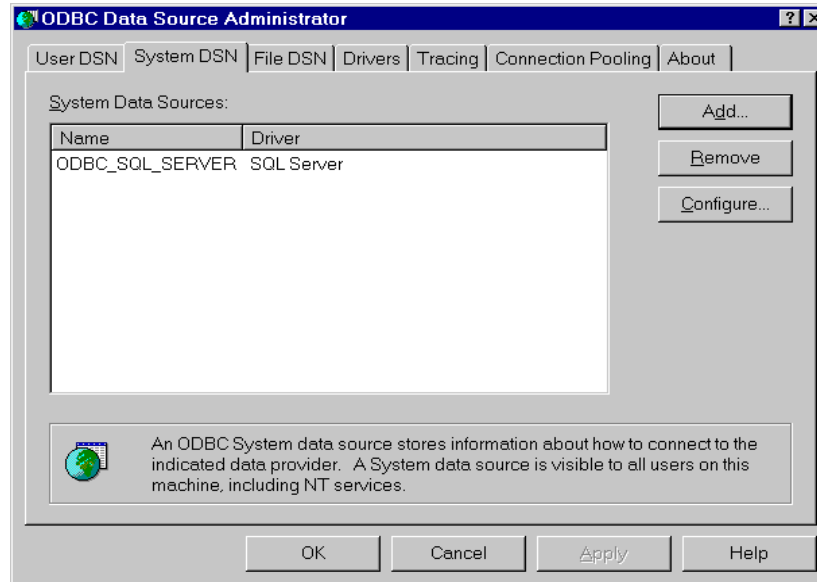
contact your database vendor for information on obtaining the appropriate 32-bit ODBC driver for your database.



Select the ODBC driver that corresponds to your database from the list presented. You will then be provided with a Setup configuration dialog. The Data Source Name is the alias that you will use to refer to the ODBC driver. The Data Source Name is also the name you must specify in the database-connect-string attribute of the g2-database-interface object. Please refer to the on-line help provided in each dialog by clicking on the Help button when you need assistance configuring your ODBC drivers.



After you have configured an ODBC data source driver alias, the data source name will be present in the Data Sources dialog box.



DNS Configuration via an RPC

Before the G2-ODBC Bridge can connect to a database, you must define a user DSN (data source name) on the computer on which `g2-odbc` is running. Typically, you do this by configuring Data Sources (ODBC), which you access from the Administrative Tools icon in the control panel.

When you click the Add button, the ODBC Data Source Administrator (ODSA) presents a list of ODBC drivers that have been registered on your computer. After you select one of the drivers and click the Finish button, the ODSA passes control to the driver. It asks for the information it needs to configure the DSN that it is creating. The information depends upon which driver you select. The requirements of the database and the authors of the driver decide what information is required.

When you are finished providing the requested information and click the button to create the new DSN, the ODBC driver passes a specially formatted string to an ODBC function, causing the configuration information to be recorded for access by ODBC clients, including `g2-odbc`. The string consists of a sequence of key/value pairs. The required keys depend on which ODBC driver you select.

For users of G2 who need the ability to create DSNs programmatically and who have the information about which key/value pairs are required for the ODBC driver they will use, the G2-ODBC Bridge provides system procedures for performing the last step of DSN creation.

Note To use this feature, you must contact the authors of the ODBC driver that you are using for the required key/value pairs. Gensym has no way of knowing what key/value pairs are required for any particular ODBC driver.

db-add-dsn

(*odbc-driver-name*: text, *system-dsn*: truth-value, *configuration*: structure, *interface*: g2-database-interface)

Adds a DSN to the computer on which g2-odbc is running.

db-modify-dsn

(*odbc-driver-name*: text, *system-dsn*: truth-value, *configuration*: structure, *interface*: g2-database-interface)

Modifies an existing DSN on the computer on which g2-odbc is running.

db-delete-dsn

(*odbc-driver-name*: text, *system-dsn*: truth-value, *dsn-name*: text, *interface*: g2-database-interface)

Deletes an existing DSN on the computer on which g2-odbc is running.

Here are the arguments to these procedures:

| Argument | Description |
|-------------------------|--|
| <i>odbc-driver-name</i> | The name you would see if you looked for the driver in the ODBC Data Source Administrator (Control Panel > Administrative Tools > Data Sources (ODBC) > Drivers). For example, to create a DSN for MS SQL Server, you would use the driver name SQL Server, which is toward the end of the list in the table on the Drivers tab. |
| <i>system-dsn</i> | Set to true if you are manipulating a system DSN; otherwise, set it to false . A system DSN is shared by all users, whereas a user DSN is user-specific. |
| <i>configuration</i> | The key/value pairs required by the particular ODBC driver you are adding or modifying (db-add-dsn and db-mod-dsn only). To add or modify a DSN, specify <i>dsn: name</i> pair in the structure. |
| <i>dsn-name</i> | The DSN name to delete (db-del-dsn only). |
| <i>interface</i> | The g2-database-interface that is connected to the G2-ODBC Bridge |

Interface Object Settings

You must specify the ODBC driver alias name in the `database-connect-string` attribute of your `g2-database-interface` object. For example, in the dialog above, the ODBC driver alias name is `ODBC_SQL_SERVER`.

Auto-Commit

By default, the bridge does *not* auto-commit transactions to the database. This means you must explicitly commit transactions, by using the `db-commit` procedure. You can also configure the connection so that it does auto-commit transactions.

To configure the connection to auto-commit transactions to the database:

- ➔ Use the `-A` option as part of the `remote-process-initialization-string` attribute of the `g2-database-interface` object.

User Names and Passwords

Most databases require a username and password to establish a connection. G2-ODBC supports two methods for supplying the database user name and password:

- Username and password can be supplied in the interface object.
- Username and password can be supplied in the ODBC driver definition.

If a database username and password are supplied in the `database-user` and `database-password` attributes of your `g2-database-interface` object, they will take precedence over the username and password supplied in the ODBC driver configuration.

If you want to utilize the username and password specified in the ODBC driver configuration, simply specify "" for the `database-user` and `database-password` attributes of your `g2-database-interface` object. You must specify a username and password in either the `g2-database-interface` or the ODBC driver configuration.

The database connection attributes of a `g2-database-interface` object might be configured as follows:

| G2 Attribute | Value |
|--------------------------------------|-------------------|
| <code>database-user</code> | "" |
| <code>database-password</code> | "" |
| <code>database-connect-string</code> | "ODBC_SQL_SERVER" |

Note For a complete description of how and when to connect to the database from the bridge, see “Configuring Connections” in the *G2 Database Bridge User’s Guide*.

Accessing Database Data

This section provides information on some aspects of accessing database data that are common to all G2-Database Bridges, and some aspects that are specific to the G2-ODBC Bridge. For more information on the aspects common to all G2-Database Bridges, see “Chapter 7, Querying the Database” in the *G2 Database Bridge User’s Guide*.

Database Character Sets and Text Conversion

The bridge will translate character sets between G2 and the database based on the command line option `-CHARSET`. This functionality is achieved via translator routines that convert G2 internal character sets into character sets utilized by database vendors.

Database Queries

Through your G2-ODBC Bridge, a G2 application can query databases for individual values or for lists, arrays or sequences of values. To perform a query through the G2-ODBC Bridge, your G2 KB must invoke one of the G2-Database Bridge API procedures for performing database queries.

Cursor Objects and Bind Variables

Every G2-Database Bridge procedure that performs a database query must reference an existing cursor object. The cursor object contains an SQL statement that defines the query. Bind variables represent values that may change in an SQL statement. You assign values to a bind variable when you define a cursor object using the `db-define-cursor` procedure. You can also change the values of a bind variable in an existing cursor object using the `db-set-cursor` procedure.

Bind variables increase the flexibility and efficiency of database operations. When you change the sequence of characters, case, spaces, or punctuation in an SQL statement, it forces the database to generate a new execution plan for the SQL statement. Generating this execution plan results in additional database processing and overhead. Using bind variables avoids this additional overhead.

Note You can only use bind variables with databases that support them. Databases designed for client/server operation generally do, for example MS SQL Server, Oracle, and Sybase. However, desktop databases such as Microsoft Access and Paradox do not support bind variables.

You should specify bind variables using a text-list. If you prefer, you can specify bind values as a text string (with each value separated by a comma), but this approach is not as efficient as providing a text-list. The symbol to denote a bind variable is different depending on the type of database. When you use ODBC to access a database, the bind variable is always represented by a question mark (?).

For example, the following code modifies the bind variables of the SQL statement in the cursor object `my-cursor-obj`. It initially sets the bind variable to 557 using the `db-define-cursor` procedure and then resets it to 778 using the `db-set-cursor` procedure.

```

status: symbol;
code: integer;
msg: text;
cursor-obj-name: symbol = the symbol MY-CURSOR-OBJ;
cursor-obj: class db-cursor-object;
sql-stmt: text = "select ename from emp where empid = ?";
bind-vars1, bind-vars2: class text-list;

insert "557" at the end of bind-vars1;
sql-obj,status,code,msg = call db-define-cursor(cursor-obj-name, sql-stmt,
    bind-vars1, myIO);
insert "778" at the end of bind-vars2;
status,code,msg, = call db-set-cursor(cursor-obj, bind-vars2, myIO);

```

Stored Procedures Returning Record Sets

Stored procedures in MS SQL Server can return record sets. The G2-ODBC Bridge can retrieve a record set by executing a stored procedure. Once you have a cursor to the record set, you can use all the different forms of the `db-fetch-` procedures on the cursor to retrieve data from the record set.

To support this feature, the second parameter to `db-define-cursor` accepts EXEC statements, as well as SELECT statements. The EXEC statement is used to start a stored procedure. If a cursor contains an EXEC statement and the stored procedure returns a record set, fetching on the cursor behaves just as it would if the cursor had contained an equivalent SELECT statement.

Event Notification through Triggers

Triggers provide the capability for G2 to respond to events occurring in the database. Triggers are defined by the database administrator on specific database objects such as tables. Trigger messages are conveyed to G2 via two methods:

- Messages to a G2 procedure.
- Updates to a trigger-object.

Up to 50 triggers may be set per bridge process. Please refer to the *G2 Database Bridge User's Guide* for details on how to set triggers.

ODBC does not provide a means for exporting trigger events from the database to an external process. Because of this limitation, the G2-ODBC Bridge provides a solution for implementing triggers using a trigger table and a database stored procedure. This solution utilizes a polling approach to check for trigger events which are stored in a special trigger table called `g2_triggers_tbl`. For example:

| trig_id numeric identity | trig_name varchar(30) | trig_datetime datetime | trig_msg varchar(200) |
|---|--|---|--|
| 1 | TEST_TRIG1 | Jan 1 1997 12:00AM | Trigger 1 has fired |
| 2 | TEST_TRIG1 | Jan 1 1997 12:10AM | Trigger 1 has fired |
| 3 | TEST_TRIG2 | Jan 1 1997 12:30AM | Trigger2 has fired |

Each time a trigger fires that you want G2 to monitor, a database procedure is invoked that inserts a row into the `g2_trigger_tbl` table. The bridge continually monitors the table by calling another database procedure `g2_return_trigger` to select a row from `g2_trigger_tbl` based on the trigger name supplied by G2. If the procedure finds a match, it sends the message, supplied by your trigger, and the timestamp back to G2, and deletes the row from the database table.

Gensym recommends using one of the following techniques for monitoring trigger events with the bridge:

- Run one bridge process with one or more interface objects connected to the ODBC database. If using more than one `g2-database` interface object, set up one interface object to handle triggers exclusively.
- Run more than one bridge process. Set up one bridge to handle only the triggers. Run separate bridge processes to handle all other SQL transactions.

Trigger Installation

To monitor triggers with the bridge, the SQL script file `inst_g2trigs.sql` must be run to create the trigger table and database stored procedures. The following database objects will be created:

| Object | Type | Function |
|--------------------------------|-----------|---|
| <code>g2_triggers_tbl</code> | table | Stores executed trigger messages and time stamps. |
| <code>g2_odbc_ins_trigs</code> | procedure | Inserts entries into the trigger table <code>g2_triggers_tbl</code> . |
| <code>g2odbc_proc_trigs</code> | procedure | Returns the trigger to G2 and removes the entry from the trigger table <code>g2_triggers_tbl</code> . |

To use this mechanism, any triggers that you want to monitor with the bridge must be modified to call the database procedure `g2_odbc_ins_trigs` at the end of the trigger. The database procedure takes two arguments, the name of the trigger and a message up to 200 characters in length.

Note The timestamp field is filled in automatically with the current date and time.

For example, to modify a user-defined trigger named `test_trig1` defined in MS SQL Server, the following lines in boldface must be added to the trigger:

```
CREATE TRIGGER test_trig1 ON authors FOR update
AS
DECLARE @retcode int
IF EXISTS (SELECT id FROM sysobjects WHERE name = 'g2_triggers_
tbl')
EXECUTE @retcode = g2_insert_trig_proc "test_trig1", "Trigger 1
has fired"
```

Trigger Deinstallation

To delete the trigger table and database procedures, execute the script named `deinst_g2odbc_trigs.sql`.

You can only use this script with MS SQL Server and ODBC.

Column Attribute Naming and Aliases

ODBC supports column name aliases in queries. The alias is specified in your select list after the column name and before the comma. You may specify the alias in single quotes or without quotes:

```
"select lname employee, fname, phone from customer"
```

For this example, the value from the database column `lname` will be put into a query object column attribute named `employee`, and the value from the database columns `fname` and `phone` will be put into query object column attributes named `fname` and `phone` respectively. For more information on using column aliases, reference your ODBC documentation.

A database column must not conflict with G2 reserved words. If there are conflicts, you can use aliases or database views to redefine your result table column names.

When performing SQL functions such as *max*, *min*, *sum*, *avg*, *count*, etc. from within queries, using query objects, the bridge will return the result values to query object column attributes of the same name as the column having the function performed on it, or you may specify a column alias. Be sure the column attribute in the query object is defined to the proper data type for the result value.

Using the G2-ODBC Bridge with Stored Procedures

The G2-ODBC Bridge supports two different techniques for working with stored procedures. You can only use these methods with databases and ODBC drivers that support stored procedures. Some databases, such as Microsoft Access, do not provide this support.

The first technique is to use `db-exec-stored-proc` to invoke and possibly pass parameters to a stored procedure. The stored procedure must not attempt to return values. If you need to return values to G2, you can use `db-exec-stored-proc-return`, or the stored procedure can write the results to a database table, which G2 can subsequently query.

When using `db-exec-stored-proc-return`, you must write a stored procedure named `sp_handler` that acts as a dispatcher. G2 passes two strings to `sp_handler`. Typically, the first string is the name of another stored procedure. The arguments required by the stored procedure are packed together into the second string. The `sp_handler` procedure is responsible for parsing the second string and extracting the parameters, then passing them to the specified stored procedure. If the stored procedure returns values, `sp_handler` should pack them into a single string, which it then returns to G2.

Writing an additional stored procedure, `sp_handler`, provides great flexibility for executing stored procedures with return values. This procedure can extract the input parameters from the input text, convert them to the required types, deliver them to the stored procedure being invoked, receive the appropriate number of return values, and pack them into the text string that is returned to G2.

These techniques of working with stored procedures are described in the *G2 Database Bridge User's Guide*. However, the example that appears in the guide is not specific for databases that use the ODBC driver. We now provide an example that uses MS SQL Server in [Example: Using db-exec-stored-proc](#).

If you do not need the flexibility provided by `db-exec-stored-proc-return` and can write your stored procedures to accept a single text string as input and return a single string as output, you can create a version of `sp_handler` that allows the stored procedure to be invoked directly. See [Example: Using db-exec-stored-proc-return](#).

Microsoft SQL Server is one of the most commonly used database management systems with the G2-ODBC Bridge, which provides a sample database named `pubs`. This database stores data for a hypothetical publishing house. The examples that follow use this sample database.

Example: Using db-exec-stored-proc

The following script creates a stored procedure named `g2_mssql_example1` that creates a table of the authors whose books were sold on a specified date and the total sales of the associated books.

```
IF EXISTS (SELECT id FROM sysobjects WHERE name = 'g2_mssql_example1')
    DROP PROCEDURE g2_mssql_example1
GO
CREATE PROCEDURE g2_mssql_example1 @seldate VARCHAR(10) AS
BEGIN
    SET NOCOUNT ON
    SET DATEFORMAT mdy
    IF EXISTS (SELECT id FROM sysobjects WHERE name = 'SalesByAuthor')
        DROP TABLE SalesByAuthor
    SELECT au_lname as 'Last Name', au_fname as 'First Name',
           sum(pubs.dbo.sales.qty*pubs.dbo.titles.price) as
           'Total Sales'
    INTO SalesByAuthor
    FROM pubs.dbo.authors, pubs.dbo.titleauthor,
         pubs.dbo.sales, pubs.dbo.titles
    WHERE ord_date = @seldate AND
           pubs.dbo.sales.title_id = pubs.dbo.titles.title_id AND
           pubs.dbo.authors.au_id = pubs.dbo.titleauthor.au_id AND
           pubs.dbo.titleauthor.title_id = pubs.dbo.titles.title_id
    GROUP BY au_lname, au_fname
    ORDER BY [Total Sales] Desc
END
GO

GRANT EXECUTE ON g2_mssql_example1 TO PUBLIC
GO
```

Note the `SET NOCOUNT ON` line at the start of the stored procedure. If you use MS SQL Server, you should start all stored procedures with this line to disabled the display of the number of lines selected or affected by SQL commands. This feature is useful when working interactively with the database but can result in the silent failure of stored procedures. Even in those cases where it does not cause a stored procedure to fail, it will significantly slow down processing.

The next step is to invoke the stored procedure from G2. When you invoke stored procedures in MS SQL Server, you should either set the auto-commit property of the `g2-database-interface` by setting `remote-process-initialization-string` to `-A` or you should follow the `db-exec-stored-proc` call with `db_commit`.

The following G2 procedure causes `g2_mssql_example1` to create a table named `SalesByAuthor` containing the sales information for September 14, 1994.

```
Example1()  
s : symbol ; c : integer ; m : text ;  
cmd : text = "call g2_mssql_example1('09/14/1994')"  
begin  
    s, c, m = call db-exec-stored-proc(cmd, dbio);  
    post "[s], [c], [m]";  
    s, c, m = call db-commit(dbio);  
end
```

Your procedure could build the text command so that it could pass different parameters to the stored procedure. For example:

```
cmd = "call g2_mssql_example1('[the current month]/  
[the current day of the month]/[the current year]')"
```

Note that the command string uses `call` to invoke the stored procedure, not `exec`, because ODBC uses `call` to invoke a stored procedure.

Example: Using db-exec-stored-proc-return

If you can write your stored procedures to accept a single text string as input and return a single string as output, you can use the following MS SQL Server script to create a version of `sp_handler` that allows you to use `db-exec-stored-proc-return` as if it were directly invoking the specified stored procedure.

```
IF EXISTS (SELECT id FROM sysobjects WHERE name = 'sp_handler')
    DROP PROCEDURE sp_handler
GO

CREATE PROCEDURE sp_handler @procName VARCHAR
                           @inpStr   VARCHAR
                           @outStr   VARCHAR OUT AS

BEGIN
    SET NOCOUNT ON
    IF EXISTS (SELECT id FROM sysobjects WHERE name = @procName
              AND type = 'P')
        EXEC @procName @inpStr, @outStr OUT
    ELSE
        RAISERROR ('Stored procedure not found', 12, 1)

END
GO
```

The following stored procedure, `SalesByStore`, accepts a store number as input and returns the value of that store's sales on file:

```
CREATE PROCEDURE SalesByStore @Args VARCHAR(32),
                              @RetV VARCHAR(32) OUT AS

DECLARE @StoreNr INT
DECLARE @vbk     FLOAT
BEGIN
    SET NOCOUNT ON
    SELECT @StoreNr = Cast(@Args AS INT)
    SELECT @vbk = SUM(qty*price) FROM pubs.dbo.sales, pubs.dbo.titles
        WHERE stor_id=@StoreNr AND
              pubs.dbo.sales.title_id=pubs.dbo.titles.title_id
    SELECT @RetV = CAST(@vbk AS VARCHAR(10))
END
GO
```

If you install the simple `sp_handler` shown above, you can call `SalesByStore` with a `G2` statement such as this:

```
retval, s, c, m = call db-exec-stored-proc-return("SalesByStore", "7066").
```

If the call succeeds, `retval` will contain the sales total as text.

@ A B C D E F G H I J K L M
 # N O P Q R S T U V W X Y Z

A

accessing data
 bind variables
 character sets and text conversion
 cursor objects
 database queries
 stored procedure support
 auto-commit behavior
 no is the default
 overriding the default

B

bind variables
 bug fixes
 getting information about

C

character sets
 column attributes
 aliasing
 naming
 compatibility
 G2
 G2 Gateway
 ODBC
 connecting to the database
 cursor objects
 for accessing data
 customer support services

D

Data Source Name *See* DSN
 database
 accessing data
 bind variables
 character sets and text conversion
 cursor objects
 stored procedure support
 queries

distribution materials
 driver configuration
 user names
 driver configuration passwords
 drivers
 32-bit
 conformance levels
 DSN

E

event notification
 using triggers

G

G2-ODBC Bridge
 Version 7.1 Rev. 0
 Version 7.1 Rev. 1
 Version 8.2 Rev. 0
 Version 8.3 Rev. 0
 using with stored procedures

I

interface object
 passwords
 settings
 user names

O

ODBC
 driver manager

P

password
 interface object

S

SQL

stored procedure support

stored procedures, using with G2-ODBC

Bridge

Structured Query Language

system requirements

T

technical bulletins

viewing

dynamic

text conversion

third-party software requirements

triggers

deinstallation

for event notification

installation

U

user names

interface object

V

viewing

technical bulletins

dynamic