GoldenGate Database Connector

First Steps

For Macromedia Authorware®

Version 1.0

Table of Contents

Introduction

Thank you for using GoldenGate Database Connector!

This First Steps manual is a quick overview of GoldenGate's most common features. It uses the Logon sample Authorware piece provided at http://www.GGdbc.com/getting-started-aw/ to illustrate:

- How an Authorware piece opens a connection to a GoldenGate server;
- How to search data;
- How to retrieve data;
- How to modify data.

In these First Steps, you will use your own Authorware piece and connect to Integration New Media's GoldenGate and MS SQL Servers through the Internet.

Most of all, you will realize that all of this requires very little scripting in Authorware and *no scripting at all* on the server side.

Before you Start

Before you start, make sure that you have:

- Authorware 4 or later, for MacOS or Windows;
- The Logon sample Authorware piece (available from http://www.GGdbc.com/getting-started-aw/);
- The GoldenGate Client Xtra for Auhtorware (available from http://www.GGdbc.com), and that it is located in Auhtorware's Xtras folder;
- An Internet connection.

Firewall considerations

If you are using these First Steps with a GoldenGate Server installed on your own computer or local area network, you don't need to worry about firewall issues.

It you want to connect to the GoldenGate Server located at Integration New Media *and* if you are behind a firewall *and* if your firewall is configured to disallow outbound queries, ask your system administrator to allow outbound queries from your computer so your GoldenGate client can contact the GoldenGate Server located at Integration New Media

You don't need to worry about firewall issues in all other situations.

Explore the landscape

First, run the Logon piece and make sure it works properly on your computer.

and make sure it works properly on your computer.

- Create a new account.
- Try to logon using your newly created nickname and password.
- Keep your nickname and password on hand as you will need them in the next few steps.

Initializing GoldenGate

Logon.a?p stands for Logon.a4p, Logon.a5p or Logon.a6p depending on your version of Authorware (version 4, 5 or 6).

If both the server and the client are running on your own computer, use "localhost" as the Server name instead of "GG.IntegrationNewMedia.com".

1. Open the Logon.a?p file.
2. Open the "Initialize GoldenGate" calc icon (in red).
3. Opening a connection to the remote database is a three step process:
   a. Open a connection to the GoldenGate Server:

```
GGcx := NewObject("GGConnection",
 "GG.IntegrationNewMedia.com", 1729,
 "AnyUser", "top secret")
```

   where:

   `GGConnection` is the GoldenGate connection Xtra's name;

   `GG.IntegrationNewMedia.com` is the GoldenGate Server's name (hosted at Integration New Media);

   `1729` is the GoldenGate Server's TCP/IP port;

   `AnyUser` is the name under which your project will identify itself when connecting to the GoldenGate Server

   `top secret` is AnyUser's password.

   b. Open a database instance:

```
GGdb := NewObject("GGdbe", GGcx, FirstSteps)
```

   where:

   `GGDBE` is the GoldenGate database Xtra's name;

   `gCX` is the connection instance created above;

   `FirstSteps` is the GoldenGate database connector's name. It resides on the `GG.IntegrationNewMedia.com` server.

   c. Open a table instance:

```
GGrs := NewObject("GGRecordSet", GGdb, "Users")
```

   where:

   `GGRecordSet` is the GoldenGate recordset Xtra's name;

   `gDB` is the database instance created above;

   `Users` is the name of the table required by this project. It resides in the database identified as FirstSteps.

The rest of the movie will only use the `gT` global variable to search, retrieve and store data. `gDB` and `gCX` will not be directly used.

Searching Data

Although GoldenGate accepts SQL statements for advanced data searching, knowledge of SQL is not required in most projects thanks to GoldenGate's simplified API.

In Logon.a?p, data is searched in two different situations.

- When granting access to a nickname/password combination: to identify a nickname's record and make sure the user supplied password matches the one stored in that record. This is done in the `Logon using my nickname and password` map, in the `Logon` calc icon.
- When creating a new account: to make sure a nickname is unique before adding it to the database. This is done in the `Create a new account` map, in the `Is nickname unique?` calc icon.

SetCriteria / Select

GoldenGate is also able to perform searches on other types of data fields (numerals, dates, etc.) with a wide range of comparison operators (greater, smaller, within range, full-text, etc.) See the GoldenGate Developer's Manual for details.

Searching data is performed in two steps in GoldenGate.

- Call `SetCriteria` with one or more search criteria, as in:
  ```
  CallObject(GGrs, "SetCriteria", ["Nickname", "=", "nick"])
  ```
- Call `Select` to actually trigger the search process on the server, as in:
  ```
  CallObject(GGrs, "Select")
  ```

Then you are ready to check the number of records selected (with `SelectCount`), browse through the selected records (with `GoNext`, `GoPrevious`, `GoFirst`, `GoLast` and `Go`), retrieve data (with `GetField`), etc.

The `Is nickname unique?` calc icon

This is the logic behind the `Is nickname unique?` calc icon (in the map `Create a new account`).

1. Tell GoldenGate to search for all records who's Nickname field equals the user's entry:
   ```
   CallObject(GGrs, "SetCriteria", ["Nickname", "=", EnteredNickName])
   ```
2. Trigger the search
   ```
   CallObject(GGrs, "Select")
   ```
3. Ask GoldenGate how many records were found.
   ```
   n := CallObject(GGrs, "SelectCount")
   ```
4. If one or more records were found (i.e., the nickname already exists), tell the user to pick another nickname.
   ```
   if (n > 0) then
        ErrorMessage:="This nickname already exists in the
          database. Please choose another one."
        GoTo(IconID@"Show Message and Retry")
   end if
   ```

Retrieving Data

In the Logon.a?p file, data is retrieved from the remote database for a few different reasons, such as:

- matching a nickname's password against the one supplied by the user, or
- retrieving the user's first name, last logon date and number of logon's to display on the Welcome screen.

In this section, you will learn how to retrieve the user's last name from the database and display it on the Welcome screen.

GetField

Data retrieval is primarily performed by the `GetField` method as in this excerpt from the `Logon` calc icon (in map `Logon using my nickname and password`):

```
fName := CallObject(GGrs, "GetField", "FirstName")
```

This statement retrieves the content of the field `FirstName` of the current record in table GGrs (which refers to table `Users`, as shown in *Initializing GoldenGate*, 3c).

Other data retrieval methods are `GetSelection` and `GetMedia`. See the GoldenGate Developer's Manual for details.

Retrieving the user's last name

1. To modify the `Logon` calc icon and make it display the user's last name as well as the first name and logon information, first add this statement (shown in bold):

```
lName := CallObject(GGrs, "GetField", "lastName")
fName := CallObject(GGrs, "GetField", "FirstName")
 NumberOfLogons := CallObject(GGrs, "GetField",
    "NumberOfLogons")
 LastLogon := CallObject(GGrs, "GetField",     "LastLogon", "d
MMMM yyyy hh:mm:ss")
```

As shown in *Appendix: Database Structure*, the user's last name is stored in a field named LastName, which is the parameter passed to `GetField`.

2. Then, modify the assignment to `SuccessMessage` to include (in bold):

```
SuccessMessage := "Welcome back, " ^ fName ^ lName ^ Return ^
"You successfully logged into the system." ^Return
```

3. Run the project.
4. Log on with the nickname and account you created earlier in *Explore the landscape*.

The Welcome screen will now display both your first and last name.

This same operation would have involved server-side script modification, query encoding, result parsing and/or character set translation in technologies other the GoldenGate.

Adding a New Record

In Logon.a?p, a new record is added to the remote database every time a new account is created. This is performed by the `Create new account` calc icon. This section will show you how it is implemented.

AddRecord / SetField / UpdateRecord

To add a new record with GoldenGate:

1. First call `AddRecord` to create a blank record. The new record is not saved to the database yet.
2. Then, call `SetField` as many times as needed to populate the fields of the new record.
3. Finally, call `UpdateRecord` to save the new record to the database.

The `Create new account` calc icon

This is the logic behind the `Create new account` calc icon (in map `Create a new account`).

4. Tell GoldenGate to add a blank record.

```
CallObject(GGrs, "AddRecord")
```

5. Assign user's entry to the fields of the new record.

```
CallObject(GGrs, "SetField", "FirstName", EnteredFirstName)
CallObject(GGrs, "SetField", "LastName", EnteredLastName)
CallObject(GGrs, "SetField", "NickName", EnteredNickName)
CallObject(GGrs, "SetField", "Password", EnteredPassword)
CallObject(GGrs, "SetField", "FavoriteColor",
   EnteredFavoriteColor)
CallObject(GGrs, "SetField", "Mission", EnteredMissionInLife)
```

6. Initialize the fields that do not depend on user interface. The current date and time is formatted as a property list and assigned to `DTlist` in the `Get Current DateTime` calc icon..

```
CallObject(GGrs, "SetField", "LastLogon",  DTlist)
CallObject(GGrs, "SetField", "NumberOfLogons", 1)
```

7. Save the record to the database.

```
CallObject(GGrs, "UpdateRecord")
```

It is always a good practice to initialize *all* the fields of a new record. In this case, it is not necessary to initialize the ID field because it is defined to automatically increment in the database. See database structure in *Appendix: Database Structure*.

Appendix: Database Structure

GoldenGate Database Connector also supports Oracle, MySQL and other ODBC-compliant databases. See http://www.GGdbc.com for details.

The MS SQL database used by the Logon Sample is hosted at Integration New Media.

Its structure is: