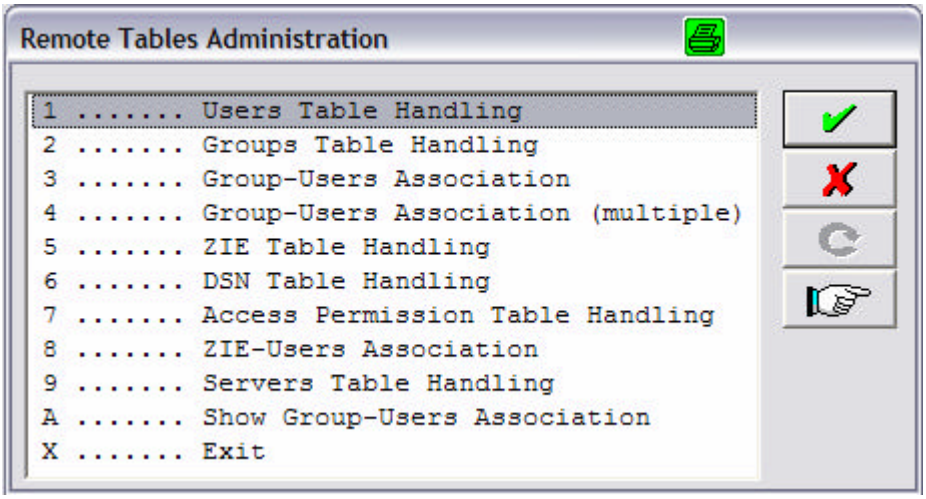
CLIENT/SERVER ADMINISTRATION TABLES

CSProX can work as a stand alone system or as a client/server system. When used as a client/server system, it requires the definition and maintenance of a number of administration tables we will describe in the following pages. Although the tables can be maintained using the same data base management the institution is using, we have provided a set of applications for remote administration developed in CSProX that are more convenient to use. These applications are packed in a special ZIE called "RENTRY.ZIE" that can be executed like any other project, provided you are an administrator and have the rights to run them.

The first step, once the server has been properly installed, is to run the client module ("CSEntry.exe") and select the "RENTRY.PFR" under the proper directory (depends where you installed the client components of CSProX. After this step has been completed, the normal connection dialog is shown and you will have to identify yourself. Note that at this point, the PFR data section of this dialog has been pre-defined with the ZIE name ("RENTRY") and the server address obtained from the "RENTRY.PFR" file. Pressing the "connect/login" button should lead to the RENTRY main menu where you can select the desired option. The menu looks as follows:



It is important to point out that although these tables can be defined in any order, it's highly advisable to follow some pre-established order to prevent inconsistencies between tables that are related to each other. Figure 1 illustrates a schema where two servers are running with two different projects having in common some project's tables. We will use this schema to describe the need for the administration tables and how links or relations exist among them. At the end of this documentation, we will give a step by step

recommendation related to the order in which we advice you to populate the various administrative tables.

The Server Table

As it can be observed on figure 1, there might be any number of CSProX servers running at any one time either in one computer or in different computers. Each of them runs with different projects belonging to the same institution or different institution sharing some resources. The way clients interact with a specific server is through the "Server Table" described in figure 2, which links the internal server name with the server address (i.e. http://server_address or https://server_address)

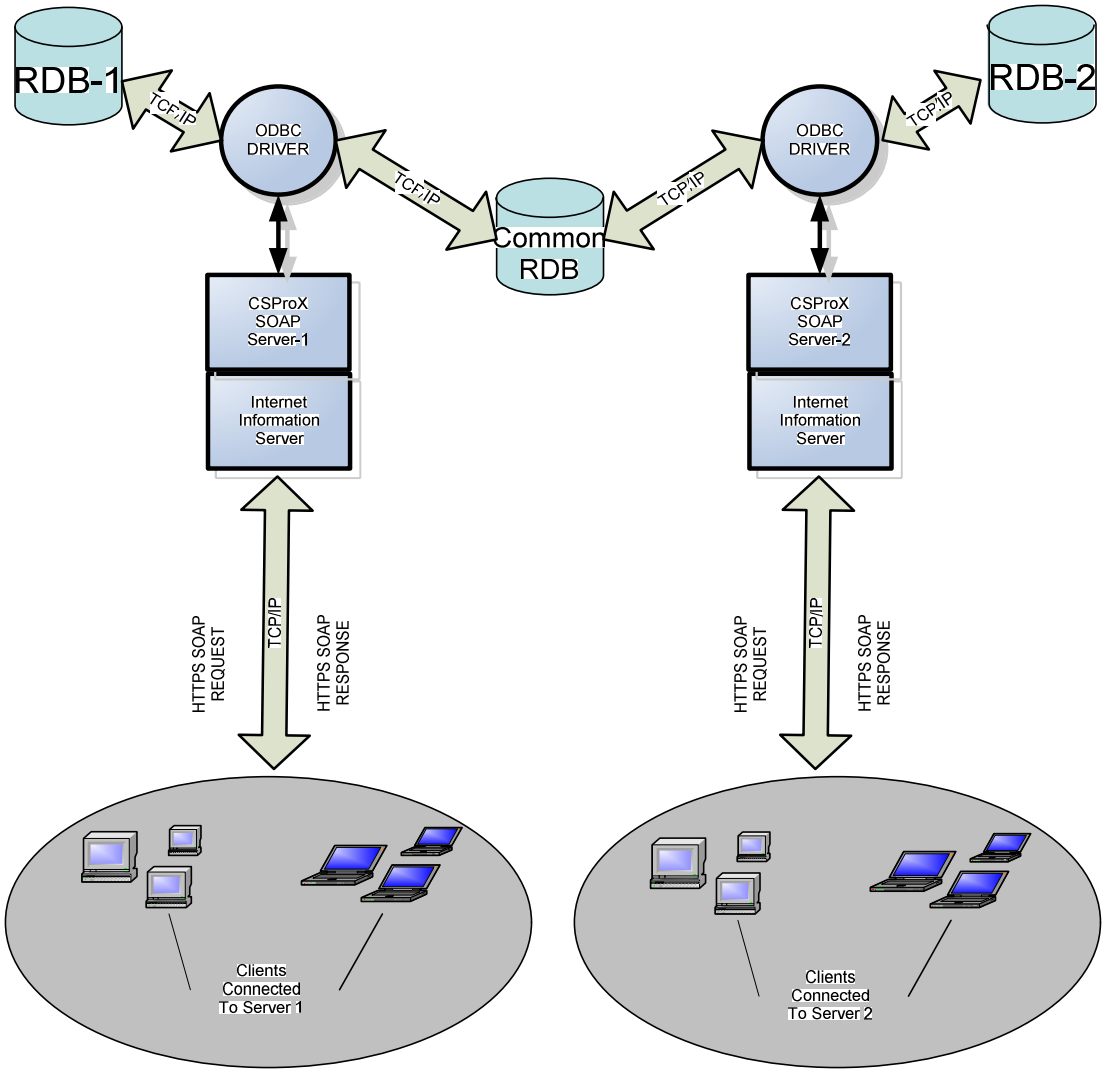


Figure 1: CSProX Schema

Link between the Server and the Project

Since you might have more than one server running with different projects, the system needs to make the association between one server and the specific project is running. This association is defined through the "Project-Name.PFR" which has been fully explained in the CSProX installation manual. Notwithstanding this, the ".PFR" file has two basic information elements: the server address and the ZIE (project) name that creates the link between them.

How the .PFR is invoked

When CSEntry is run, three different parameters can be passed to it: "App-Name.ent" "App-Name.pff" or "Project-Name.pfr". The last one tells CSEntry that it's going to work in remote mode and at the same time, specifies through the PFR file content, the server address and project/ZIE name.

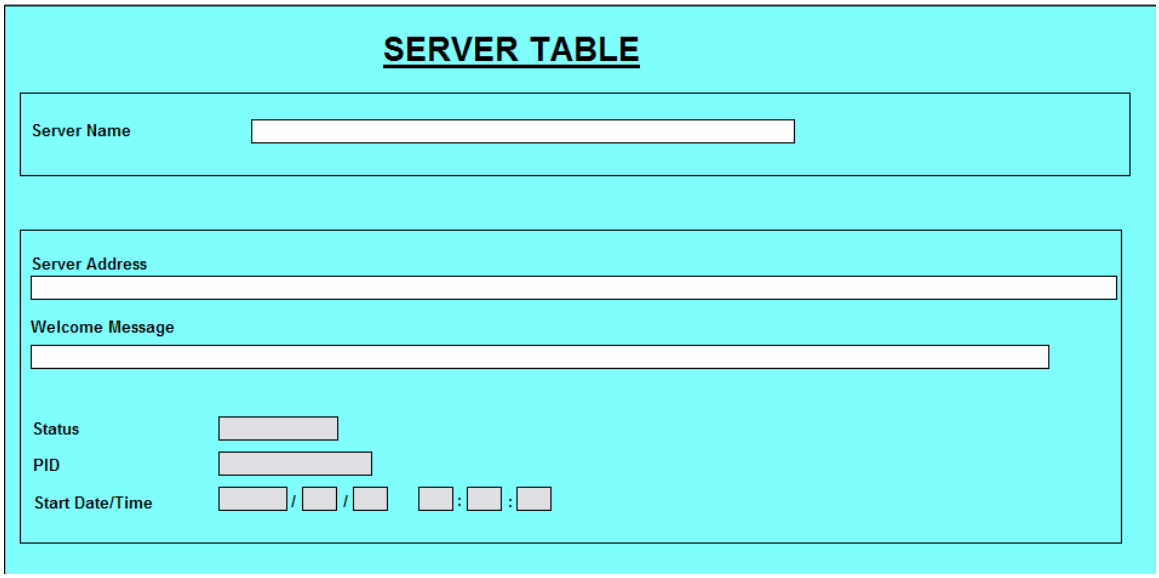


Figure 2: Server Table

The DSN Table

This table allows you to link each project's remote data dictionary to a common or more than one database according to the project administrator needs or preference.

The table also captures the database login and password that will be requested every time the administrator(s) connect to the database through the ODBC driver.

It is important to point out that the project database(s) can be entirely defined in one machine or they might have tables defined in different computers. However, if the tables reside in a different machine, other than the server, there will be some extra overhead due to the communication between the server and the machine storing the database.

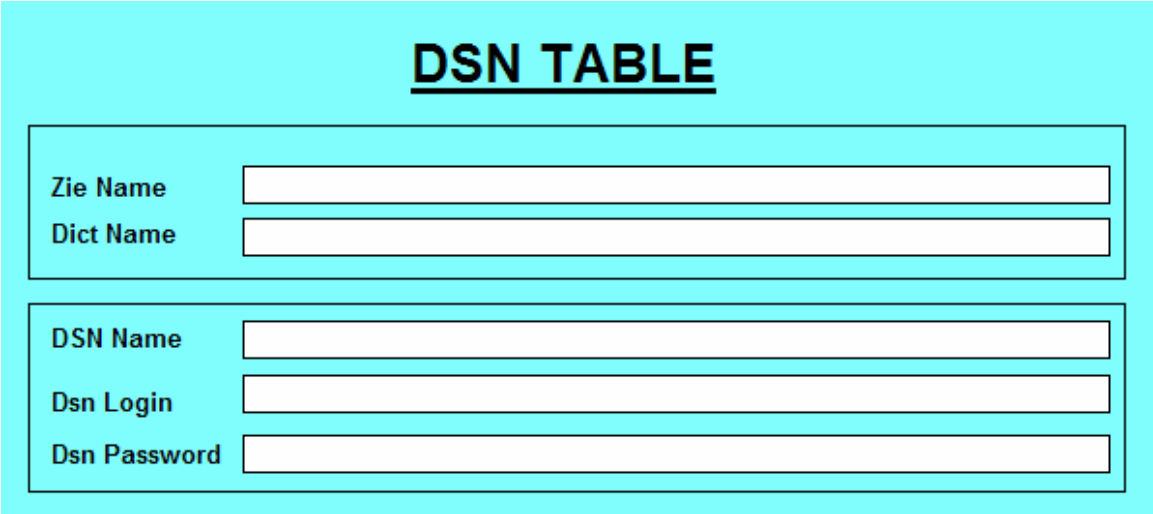


Figure 3: DSN Table

Users' Group Table

One important characteristic of the CSProX Client/Server approach is that clients can be classified in clusters or groups according to some concept(s). In turn, groups can also be clustered higher level groups creating a tree structure.

The important consequence is that clients can inherit the group characteristics (e.g. access rights to DB or dictionaries) although they can be individually altered by the project administrator.

Combining the group and client access rights, the administrator can let any client to “see” all cases entered by the group s/he belongs to, to “see” and modify them or simply to “see” and modify only the cases individually entered.

As it can be easily derived from the paragraphs above, there are two tables that are strongly related to the group table and they are the Client/User table and the Access Rights Table. We will explain that in detail in the following pages. However, it's important to bear in mind that: (1) Groups exist only as a way to facilitate the organization of the interviewers or Data Entry Operators (DEO). (2) If the number of interviewers or DEO is small and thus there is no need to group them in clusters, they still need to be assigned to a common group created by system default at the time the administration tables are generated. The name of this group is "rentry_admin_users" and it will be seen when the group table application is called from the remote tables' administration menu.

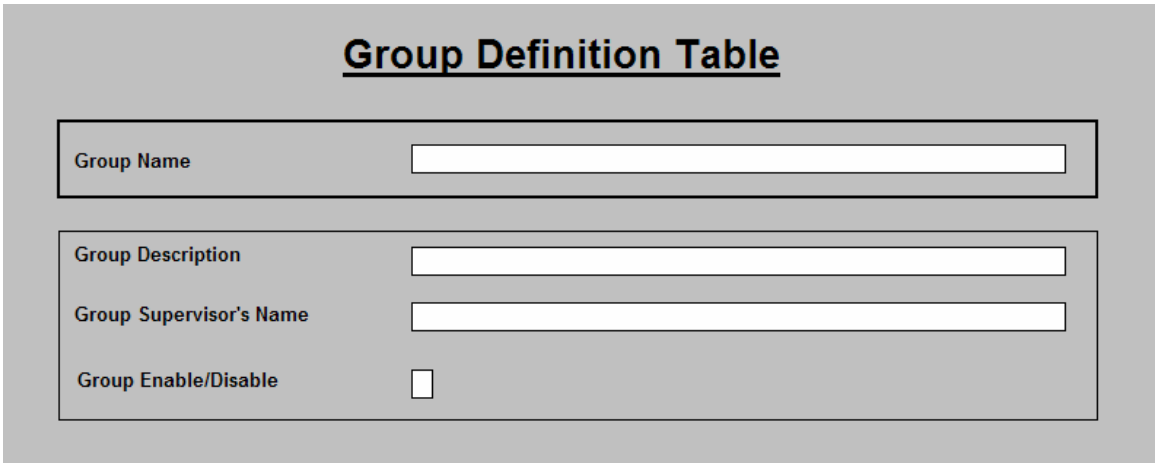


Figure 4: Group Definition Table

At the time we define the group, we are just creating the entity and we are also providing some general information attached to this group like the group description and its supervisor name. In addition, we are also stating whether the group is enable or disable. The project administrator might decide whether the group –and therefore its users- will be at this time disabled or not.

As it can be observed, at this point we are not specifying what users will be members of this group. This task will be done once all groups and users have been created individually through the "Groups-Users Association Table" explained later in this documentation.

Users' Table

Among many general pieces of information provided in this table, it does gather some very important one like: (1) The user name which will be recognized at the time of logging in as a valid user; (2) The password to validate the user's name; (3) The type of user specifying whether s/he will be an administrator or not; (4) The information specifying whether the user is enable or disable (at this time).

Once again we point out the need to associate each user to an existing group since it will be the only way to operate once we get started.

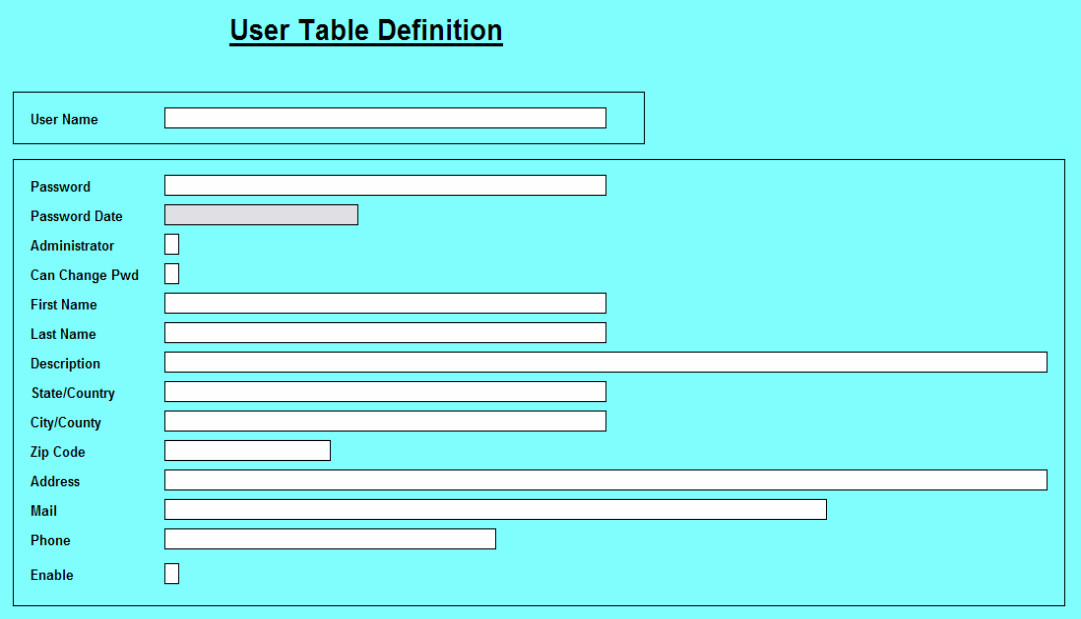


Figure 5: Users' Table Definition

For institutions with a large number of users it's advisable to have a different method to capture this information keeping the same data structure and then simply add it taking advantage of the system synchronization between the client and the server. Another possibility is simply to use the relational data base features to add them directly to the users' table.

User-Group Association Table

This table creates the link between a user or a group and the group it belongs to. Note that groups can also be members of another group (parent group) creating a tree hierarchy where the leaves would be the users.

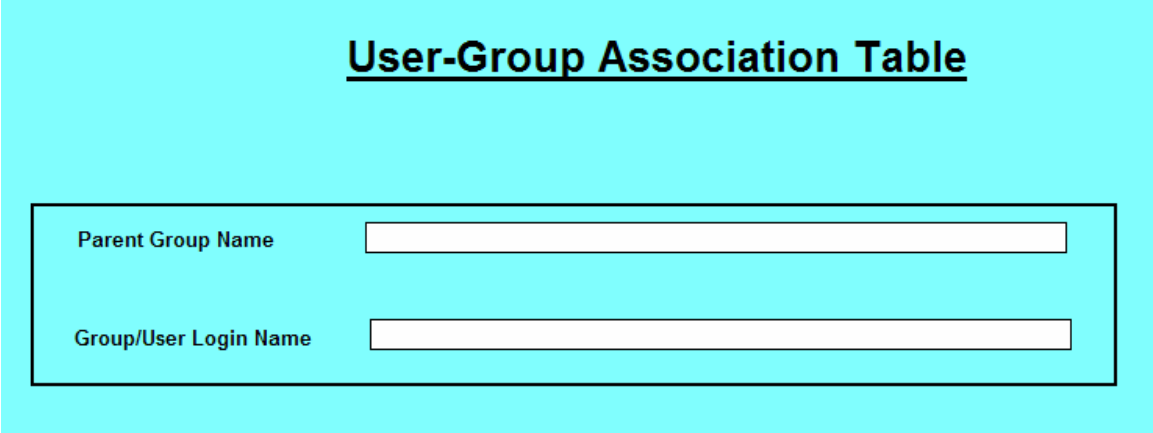


Figure 6: User-Group Association Table

The table shown above can be time consuming if your institution has a large number of users since you will need to enter one by one all the users belonging to each group. However, if you assign a common prefix to all the users belonging to the same group (i.e. G1_ to all users of Group 1 and so on), the task can be facilitated by choosing the option 4 of the "Remote Tables Administration Menu" explained in the CSProX Installation Manual.

Since this topic is important for institutions having a large number of users, we will review it in detail. Note that there are two options to define the user-group association: one –shown above- for a simple one to one association and the other, to define a set of users to a specific group. Figure 7 shows the option 4 of the menu to define simultaneously several users to group1.

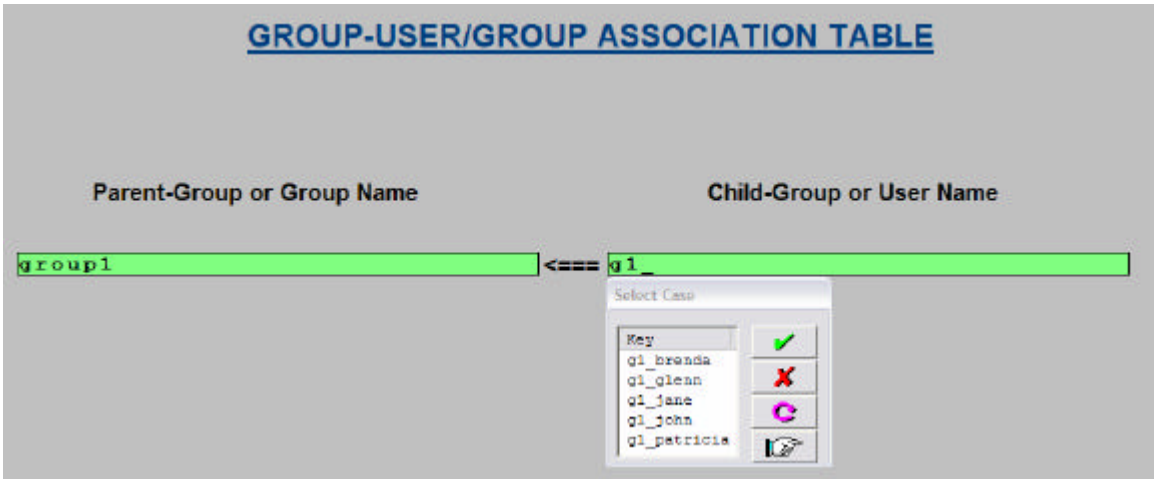


Figure 7: Group-User Association (multiple)

Analyzing figure 7, we can conclude that: (1) The administrator had previously created the group “group1”; (2) S/he had also created the users displayed on the SELCASE dialog box taking the precaution of heading the users’ name with a common prefix “g1_”; (3) By entering the prefix and pressing the <Enter> key, all users that start with this prefix are displayed; (4) The last step to assign some or all these users to “group1” is to select them using the traditional multiple selection method used in Windows (i.e. using <Ctrl> and mouse click on top of the desired entries or to select all of them, selecting the first one and then using <Shift> and mouse click on the last entry –note that <Ctrl><A> will not work-).

ZIE/Project Table

As explained in the installation documentation, any application or set of applications developed for any one project has to be converted into a ZIE file. The ZIE is then added to the ZIE table to be accessible for users (interviewers or data entry operators). As it’s shown on figure 8, besides the ZIE name, this table stores the ZIE version, a field showing whether the ZIE is enable or disable and the minimum CSProX version required by the project. This information allows the system to know when a user has an old ZIE/project version and needs to be updated to operate, operation that is automatically done by the CSProX. In a similar manner, if a project requires of a newer version of CSProX to operate correctly, when a user has an older version will be advised to update CSProX before continuing the session. All these checking’s take place at the time the user is connecting to the server and the information stored in this table is used with this purpose.

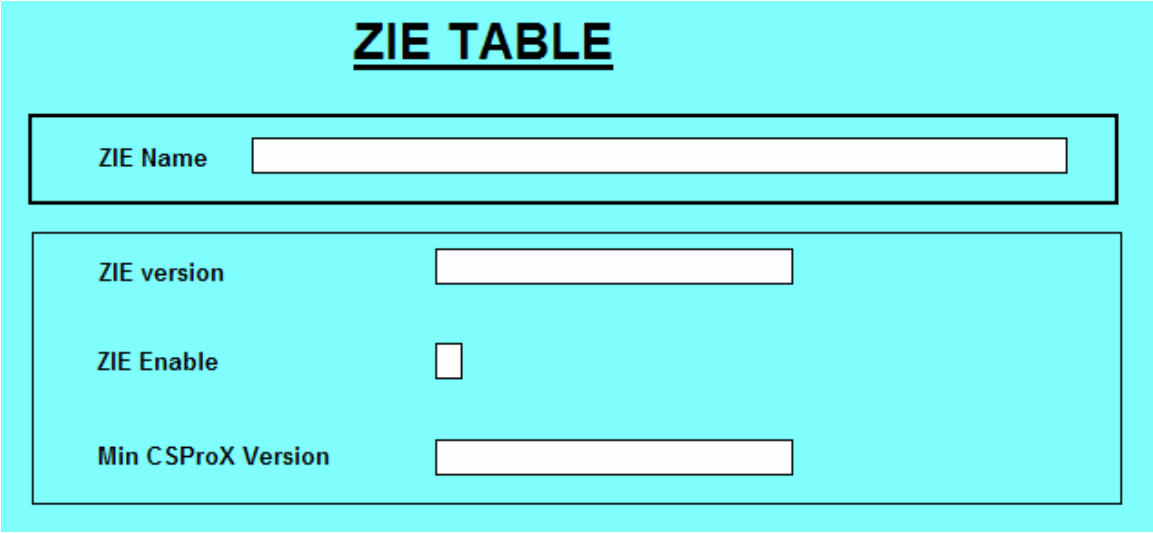


Figure 8: ZIE Table

ZIE-User Table

Since an institution might have multiple projects/ZIEs, it's necessary to specify which users have access to each specific ZIE. This table creates the user's permission to connect to a ZIE by specifying the user/group name and the ZIE name. To define the link, both the user/group and the ZIE have to be defined in their respective tables. If a group name is entered, all members of the group are granted access (or not) to the ZIE.

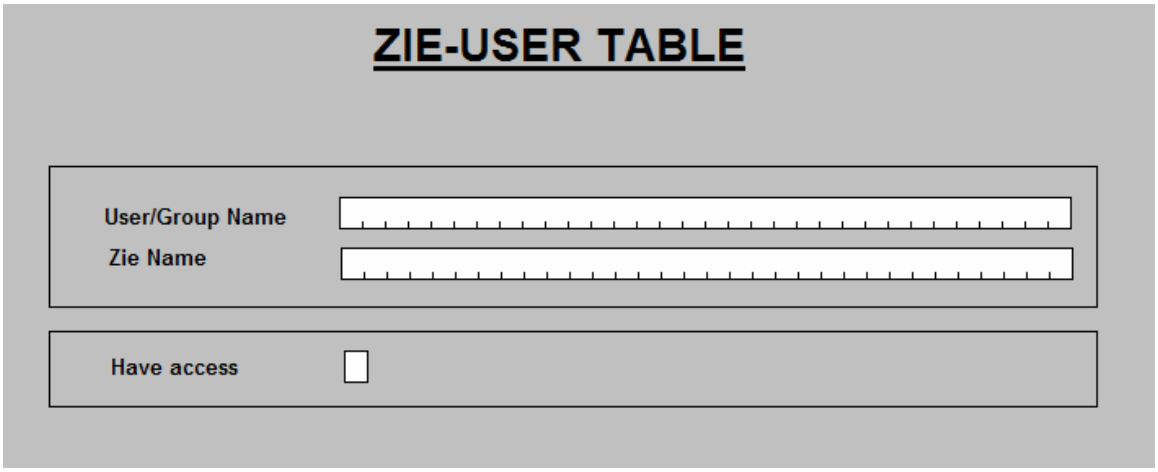


Figure 9: ZIE-User Table

Users/Group Access Permissions Table

As shown on figure 10, the access permission is granted to a group/users for a given project for a given data dictionary. Therefore, the accessibility is granted at the data dictionary level (file level) rather than at the project level (data base level). It is important to point out that only remote DD needs to have access permissions; local data dictionaries like simple lookup tables don't need to be included in this table.

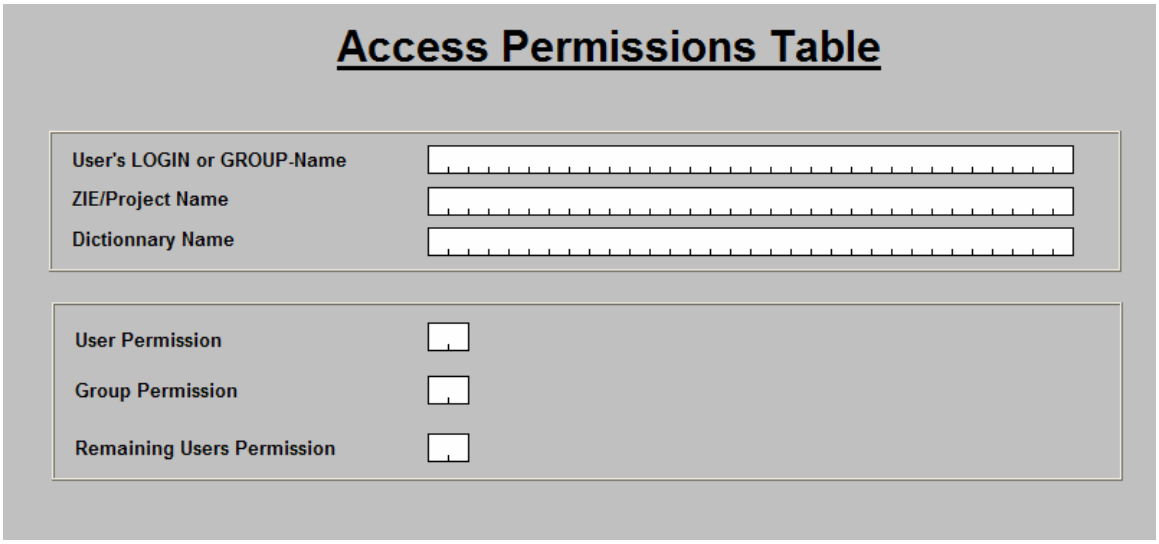


Figure 10: Access Permissions Table

If a group is specified (first field of table) then: (1) The user permission shows the accessibility of any single user regarding the data dictionary of the project specified (third and second field respectively). (2) The group permission shows the accessibility of the whole group regarding the cases written by any one user. (3) The Remaining users permission shows the accessibility that users who are members of any other group have regarding the cases written by any one user of the group specified.

The accessibility explained above is applied to all users of the group specified and can be figured as a template applied to the whole group's users. However, if we want to grant different permissions to a specific user, then a user's name needs to be specified in the first field of the table.

The permissions fields have each two characters; the first character always refers to the "Read" permission and the second to the "Write" permission (to the file associated to the data dictionary specified).

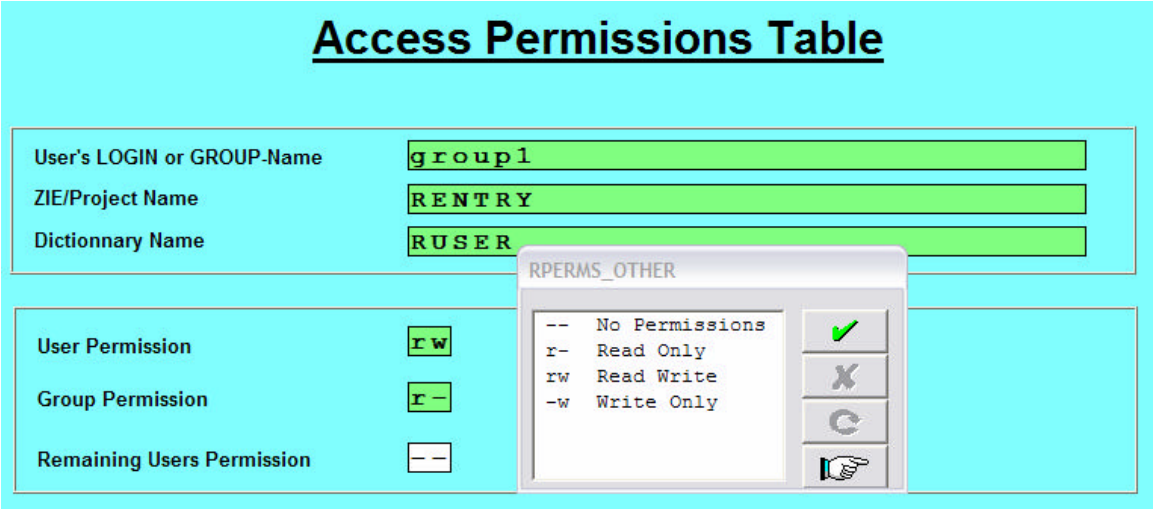


Figure 11: Permissions Example

In the example above, any user of "group1" has "Read-Write" permission to the "RUSER" table/file of the "RENTRY" (remote administration tables) project; however, all other members of this group have limited access to cases written by any one user (Read only). All members of other groups have no access to the cases written by any member of "group1"; they don't even see the case keys displayed on the left part of the screen (cases entered).

Populating the various Administration Tables

The following paragraphs describe the order in which you should populate the administration tables to ensure the consistency between them and the correct system operation. Note that the administrator can decide to populate them in any rational order and the order we suggest is just that, a simple suggestion.

1. The first step is to add to the server table the information regarding to the physical address and the internal name by which it will be known.

2. Probably, the second step should be to add the project to the ZIE table since this is the starting point to have all the applications making up the project packed and visible for everybody that will be "using" them.

3. The third step should be the definition of the DSN name for a ZIE and each of its remote data dictionaries.

4. Once we have brought the project to life, we have to let the system know who will be entitled to see and interact with this project. These set of users of the applications packed in the ZIE have to be added to the users table, the only way the system will recognize them as authorized players. Note that at this point, we are adding users without specification of the ZIE they will be using.

5. Subsequently, we can start thinking about how we will structure our users, according to what concept(s) we will group them. This concept can be related to the geographic location they will be working on, the supervisor they will be reporting to, or any other concept you might want to apply. As it has been mentioned before, groups are a useful entity to grant permissions to users instead of going user by user defining there individual data dictionaries access rights. Thus, we want to define the groups according to these concepts (Group Table) and later, make the association between each group and the corresponding users.

6. At this point we are ready to create the association between groups and their individual users according to the concepts specified above. Note that you might have decided that for your purposes, the stratification is not needed and thus you will have only one group. However, this step is still necessary in this particular case.

7. Now we are ready to specify or create the association between groups/users and the ZIE or project they will be interacting with. A given ZIE can't be accessed by any user that has not been associated by direct name or indirectly through the group s/he belongs to.

8. In our structured process, we can define next the permissions in the Access Permission Table where we decide the access type of each group/user to each remote data dictionary of the ZIE/project and, at the same time, the permissions other group(s) will have regarding the cases entered/gathered by the group in question.