Method

In order to properly follow Sybase name spacing (where tables are owned by users in databases), each user shall require its own DB2 schema in which objects owned by that user shall be created. This schema shall be named as follows:

- ***DBNAME_USERNAME*** – Where *DBNAME* is the name of the database in which the user exists and *USERNAME* is the name of the user within that database.

This schema shall be created at the time the user is added to the database and shall initially be created with CREATEIN rights revoked from the user since, by default, users cannot create objects in Sybase. When a user is granted or revoke a CREATE privilege, and then CREATE rights will be inferred to the user's schema.

Logins will be created and dropped within the security schema, the table values are as follow:

```
sp_addlogin loginame, passwd [, defdb]  [, deflanguage] [, fullname] [, passwdexp]
                                        [, minpwdlen] [, maxfailedlogins] [, auth_mech]
```

User Name Map

The SYB_SYSUSERS table shall represent the Sybase **sysusers** table. This table shall be identical to the Sybase version of the table with the exception that it shall maintain two additional columns:

- **DB2_ROLE** – A VARCHAR(128) column used for permission management. This column shall be used to determine the name of the role to which permissions shall be granted within DB2 when user permissions are being granted within Sybase.
- **DB2_SCHEMA** – A VARCHAR(255) column that is used when objects are created by the user to determine in which schema those are to be created. In the case of rows in **sysusers** that represent groups and roles this column will be NULL.

Logins

Because DB2 will be performing user authentication, **customers will be required to first define each of their Sybase users as a user in the authentication plugin that they have configured for the DB2 instance** (typically this will be using operating system authentication). Once they have performed this step, the Sybase portion of the login will be defined as per the remaining subsections below.

sp_addlogin

As with Sybase, **sp_addlogin** will be used to establish an existing DB2 login as a Sybase login. The SQL Skin version of this procedure will behave as follows:

- The **@passwd**, **@passwdexp, @minpwdlen, @maxfailedlogins, and @auth_mech parameters** will be silently ignored. Because passwords are maintained externally they cannot be set or changed nor can other password policies be affected.
- The **@loginname** parameter must match the name of the DB2 user that you are adding. Because DB2 login are defined externally (typically in the operating system) there is no reliable way to validate this, so there will be no error raised if the user attempts to add a login that does not have a corresponding operating system login (or whatever mechanism DB2 is using).
- The DB2 account will be granted CONNECT, allowing it to connect to the DB2 server.

- The DB2 account will be granted SYB_LOGIN, allow it to access SQL Skin metadata repository resources.
- For each database that has a *guest* user defined, the DB2 login will be granted the guest user's role (SYB_U_*DBNAME*_GUEST – see **sp_adduser**, below). This will infer all of the rights of guest for that database to the login.

sp_droplogin

The **sp_droplogin** procedure will perform the following activities:

- The login will be removed from the Sybase system catalogs as per the normal rules for Sybase's **sp_droplogin** (for example, it will verify that the login is no longer a valid user in any database before allowing the login to be dropped) and will raise the appropriate error messages where necessary.
- It will **not** revoke the DB2 CONNECT privilege. Since we have no way of knowing if the underlying DB2 connection is being used for activities other than ACS, it may be inappropriate for us to remove this privilege. **This means the DB2 login will remain valid and usable outside of ACS**.
- Customers should be notified that they should manually issue a REVOKE CONNECT on the login to completely disable the account. o In the future we should provide a configuration option in ACS_CONFIGURATION or ACS_OPTIONS to allow customers to modify this behavior.
- It will **not** revoke privileges granted to the DB2 login. Because we do not track privileges that where granted via SQL Skin in our Sybase system catalogs (other than role membership), and the underlying DB2 login may have been granted rights outside of ACS, we cannot know which rights are safe to revoke. **This means that the DB2 login will maintain all rights and permissions with the one exception that they will no longer be capable of logging in as a "Sybase" login".**
- Customers should be notified that they should issue a **[NOTE: IS THERE A WAY TO ASK DB2 TO REVOKE ALL PRIVELEGES FROM A USER??]**
- Future versions of SQL Skin should revoke all ACS-specific roles that have been granted to the login.

sp_password

The **sp_password** procedure will be provided in order to allow applications to execute it without errors; however it will have the following behavioral differences:

- **The current password provided will not be validated.**
- **The underlying password will not be changed.** Customers must be notified that they will need to utilize the underlying authentication mechanism themselves to change the password.

sp_modifylogin

The **sp_modifylogin** procedure shall be provided by ACS, however it will have the following notes:

- The option **authenticate with** will be silently ignored.
- The options **add default role, drop default role** will be silently ignored. Roles are always active in ACS.
- The options **passwd expiration, min passwd length,** and **max failed_logins** will be silently ignored.
- The option **login script** will be silently ignored.