

Chapter 8.1 - Setting up the SHRINE user

The Adapter uses the SHRINE user to run queries on i2b2's CRC.

To set up the standard SHRINE application user, you will need to connect to the i2b2pm database and insert entries for this user (**NB**: the actual database name may be different from what is displayed here, but the essential element is that you should locate the tables below in the correct database and schema before proceeding):

```
insert into PM_USER_DATA
(user_id, full_name, password, status_cd)
values
('shrine', 'SHRINE User', '9117d59a69dc49807671a51f10ab7f', 'A');

-- The password hash you see above is for 'demouser' . Use something else.

insert into PM_PROJECT_USER_ROLES
(PROJECT_ID, USER_ID, USER_ROLE_CD, STATUS_CD)
values
('SHRINE', 'shrine', 'USER', 'A');

insert into PM_PROJECT_USER_ROLES
(PROJECT_ID, USER_ID, USER_ROLE_CD, STATUS_CD)
values
('SHRINE', 'shrine', 'DATA_OBFSC', 'A');
```

A brief note of explanation is in order for the insert statement for the PM_USER_DATA table. The value for the password field is actually a modified MD5 hash of the actual password. However, to the best of our knowledge, a standard MD5 algorithm does not always generate the correct hash in this situation. Instead, the following python script should be used to generate the hash:

python MD5 hash generator

```
import hashlib
import string
import struct
import sys

def i2b2_pass(passwd):
    md5 = hashlib.md5()
    md5.update(passwd)
    digest = md5.digest()
    byte_list = []
    for b in digest:
        byte_list.append("%x" % struct.unpack('B', b))
    return string.join(byte_list, '')

def main():
    if len(sys.argv) != 2:
        print("usage: %s passwd" % sys.argv[0])
        sys.exit(1)
    passwd = sys.argv[1]
    print("%s" % i2b2_pass(passwd))

if __name__ == "__main__":
    main()
```

You should save this code to your local disk (for the purpose of this example, you should name it "i2b2_hash_generator.py"), set its execute bit, and then run the command below. If your syntax is correct, the program should display the correct hash that you can use to create the user as indicated above:

running python MD5 hash generator

```
$ python i2b2_hash_generator.py <insert_your_human-readable_password>
```