



# CHAPTER 17

## User Management Functions

---

This chapter provides information about the following user management functions:

- [createUser](#), page 17-1
- [deleteUser](#), page 17-2
- [getDiscoveryAuthInfo](#), page 17-3
- [listAllUsers](#), page 17-3
- [readUserInfo](#), page 17-4
- [writeUserInfo](#), page 17-4

### createUser

#### Synopsis

```
UserInfo createUser(UserToken token, String username, String password, Role role) throws  
RemoteException;
```

#### Description

This function creates a new user. The user must log in using the administrator role to have permission to invoke this function. If a role is not entered, a default role of REPORTMGR is assigned.

#### Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
username	String, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	The username.

Parameter	Type	Value	Description
password	String, mandatory	Up to 64 ASCII characters in the range from x21 to x7A	The user password.
role	Role, mandatory	Enumeration value	Role is an enumeration value for the given user.

**Return**

This function returns a UserInfo object if the operation is successful.

**Error and Exception**

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, this function returns null.

## deleteUser

**Synopsis**

```
boolean deleteUser(UserToken token, String username) throws RemoteException;
```

**Description**

This function deletes a user from the system. The user must log in using the administrator role to have permission to invoke this function.

**Input Parameters**

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
username	String, mandatory	—	The username.

**Return**

This function returns true if the operation is successful, and false if it is unsuccessful.

**Error and Exception**

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, this function returns false.

# getDiscoveryAuthInfo

## Synopsis

DiscoveryAuthInfo getDiscoveryAuthInfo (UserToken token) throws RemoteException;

## Description

This function returns a collection of usernames/passwords/enable passwords set by admin user during Discovery. The user must log in using the administrator role to have permission to invoke this function.

## Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.

## Return

The function returns a DiscoveryAuthInfo Object.

## Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

# listAllUsers

## Synopsis

String[] listAllUsers(UserToken token) throws RemoteException;

## Description

This function returns a list of usernames in the system. The user must log in using the administrator role to have permission to invoke this function.

## Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.

## Return

This function returns a string array representing the names of users.

## Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, this function returns null.

## readUserInfo

### Synopsis

```
UserInfo readUserInfo(UserToken token, String username) throws RemoteException;
```

### Description

This function retrieves user information. If token is the only input, it retrieves the user information associated with token. If both token and username are present, it retrieves the information associated with username. The latter can only be called by a user with the Administrator role.

### Input Parameters

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.

### Return

This function returns UserInfo objects.

### Error and Exception

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, this function returns null.

## writeUserInfo

### Synopsis

```
boolean writeUserInfo(UserToken token, UserInfo user) throws RemoteException;
```

### Description

This function writes the given UserInfo object into the inventory. The input UserInfo object must exist and be retrieved from the inventory by the readUserInfo() function.

**Input Parameters**

Parameter	Type	Value	Description
token	UserToken, mandatory	—	A token that represents your authorization pass, which is obtained after you invoke the login function and are authenticated by the back-end server.
user	UserInfo object, mandatory	—	UserInfo object.

**Return**

This function returns true if the operation is successful and false if it is unsuccessful.

**Error and Exception**

If a system error prevents the operation from completing, a RemoteException is thrown.

When an error occurs, this function returns false.

