

Disaster Recovery database restoration and other special procedures

This is a holder record for short "how to" sections on fixes and repairs.

Procedure for Restore following database corruption on multi-site replication-based system

1. Find a "good" version of the system you want to use to replace the corrupted db.
2. Run the following script in the DB Util of a system linked to the "good" database to create a clean backup file with all the data you need.

```
SAVE UPDATES FOR DATABASE "DATA"  
  AS Discard TO STORE Backup!  
BACKUP DATABASE "Data" AS "ForRestore" TO STORE BACKUP  
  EXCLUDE PUBLISHED UPDATES!  
COPY FILE "ForRestore.EDBBkp" IN STORE Backup  
  TO "ForRestore.EDBBkp" IN STORE ServerUploads!
```

3. Connect to the server of the corrupted database
 - Move the "ForRestore" backup file from the ServerUploads store on the "good" machine into the Backup store of the restore machine. (Note this step may require more than one copy operation, as "ServerUploads" may not be directly accessible from the restore machine.)
 - Disconnect all users from the database.
 - Open OrxServer and disconnect, then use the "restore" button select the "ForCloud" file created in the prior step. This process will empty all upload and download stores.

Procedure for Upgrade of a Server with a new database

The upgrade process usually involves a Developer taking a backup copy of the main App database and undertaking extensions and changes to it, and adding to the records in the Framework tables (BusinessObjects, Resources etc.)

The ideal upgrade incorporates all the changes made by the Developer with all the new user data added since the Developer took the backup copy. Orixia includes an automated mechanism to make this process relatively easy.

1. On the main App Server Restore the Developer's DB into a new database called "DevDB".
2. Use Orixia's built in tools to generate a SQL Transformation Script this will compare the meta data of the App Database with the DevDB and create SQL to change the App Database so its structure is the same as the DevDB.
3. Review and tidy up the Transformation Script. Backup the App Database, then run the Transformation Script.
4. The App Database will now match the DevDB in structure, but no changes made by the Developer to the Framework data-tables will yet have been applied. To add these, Restore the main App Database using the Developer's backup, but **only restore** the Framework data-tables. In this way all the Developer's new framework additions will be added, but all the user's data tables will be untouched.

To Generate a "Difference" SQL Script between two databases

Run the following two SQL Scripts to first compare the two databases, and then to retrieve the SQL which can be used for the Transformation Script.

```
COMPARE DATABASE "[SourceDatabaseName]"  
TO "[TargetDatabaseName]"
```

```
SELECT LIST(AlterSQL, '!', + #13 + #13) as AlterScript FROM Information.SchemaDifference
```

Copy the resulting "Alter Script" record into an editor window for editing and saving. It should be run in the Database Management Utility, with all other users disconnected from the database after a backup of the main database has been made.

SQL To Restore "just" the Framework data-tables

```
RESTORE DATABASE "Data" FROM "[backupfilename]" IN STORE "[storename]"  
TABLES BusinessObjects, Resources, Searches!
```