Team Development Database Considerations - Shared Versus Local

Team Development Database Considerations

When developing database applications in a team development environment, considerations must be made on the development database environment. Servoy Developer allows a developer to connect to the database they are developing against. This database can be a local database to the development machine, or it could be a shared database on a the database server.

Reasons to Use a Local Database

- Distributed developers - Developers may be in various locations and a shared database may not be conducive to security, maintenance, etc.
- Offline development - Developer may not have consistent connections to the database server.
- Platform preferences - Developers may have different database platforms or may purposely need to use a different platform for various reasons.
- Development isolation - If a single developer is working on a new features that may change the database in a way that will affect other developers adversely.

Reasons to Use a Shared Database

- Test data - It is easier to provide test data for all developers and to use the same test data if the database is shared. Also, if test data is updated regularly from a production server, it would be easier to provide changes to development with a shared database.
- Outside database administration - if a non Servoy developer is responsible for the database structure during development, they do not even need to use Servoy to create the database structure when sharing the database. Changes will be picked up when Servoy Developer starts.
- Maintenance - easier to maintain, tune, and optimize a single database server than many.
- Security Policies - may require data not leave the office via laptops, etc.

Local Database Considerations

- Updating database structure changes - tools in Servoy Developer will help with keeping all the developers updated using DBI files that can be synchronized between developers using team sharing providers. After any team provider synchronizations, there is an additional step to apply any changes to the local developer's database, as actual changes to the database are not initialized by the team provider.
- Application metadata/test data - The actual data in a database is not synchronized through any actions in the team provider or Servoy Developer. If you are created applications that have metadata associated with their use (example: security settings, reporting frameworks, etc.), or if you are using specific test data, these changes must be dealt with by using database client tools, basic import/export functionality of Servoy, or a custom built interface in the application.

Shared Database Considerations

- Servoy sequences - If you are using Servoy sequences for primary keys (this is the default for Servoy Developer, so if you are not sure, most likely you are using Servoy sequences) and other autoenter values for a database column, then using a shared database for development can be problematic. Sequences are managed by the application server, in the case of multiple developers, there are multiple application server processes running, so each developer is trying to manage the sequence. This will cause errors when adding records to the shared database. With a shared database, instead of Servoy sequences, you should allow the database to manage the sequence and mark the key accordingly in the table editor (db identity).