

# Contents

- [Primary Key Management](#)
    - [Database specific](#)
      - [Microsoft SQL Server](#)
      - [mySQL](#)
      - [PostGreSQL](#)
      - [Oracle](#)
    - [Generic](#)
      - [Numeric Primary Keys](#)
      - [UUID Primary Keys](#)
      - [GUID Primary Keys](#)
    - [Manual](#)
- 

## Primary Key Management

Transfer has several ways of handling primary keys:

### Database specific

#### Microsoft SQL Server

Retrieve the last created Identity value for the table, e.g.

```
<id name="userid" type="numeric" />
```

#### mySQL

Retrieve the last created auto increment value for the table, e.g.

```
<id name="userid" type="numeric" />
```

#### PostGreSQL

Retrieve the last sequence value mapped to the table, e.g.

```
<object name="User" sequence="user_sequence">  
<id name="idbasic" type="numeric" />  
...
```

#### Oracle

Retrieve the last sequence value mapped to the table, e.g.

```
<object name="User" sequence="user_sequence">
<id name="idbasic" type="numeric" />
...

```

## Generic

Transfer also has the capability to generate for you:

### Numeric Primary Keys

Transfer can generate unique numeric keys per table, by setting generate='true' when the type is 'numeric' e.g.

```
<id name="userid" type="numeric" generate="true" />

```

### UUID Primary Keys

Transfer can generate UUID keys, by setting generate='true' when the type is 'UUID' e.g.

```
<id name="userid" type="UUID" generate="true" />

```

### GUID Primary Keys

Transfer can generate GUID keys, by setting generate='true' when the type is 'GUID' e.g.

```
<id name="userid" type="GUID" generate="true" />

```

## Manual

Lastly, this can all be manually overwritten simply by setting the primary key value via the set() method prior to the TransferObject being created in the database, e.g.

```
user = getTransfer().new("User");<br /> user.setUserID(1);
getTransfer().save(user);

```

This will attempt to insert the User record into the database with a ID of '1'.

Categories: • [Configuration](#)