



Jürgen Aurisch  
Markus Stausberg  
Wolfgang Auer  
SAP AG

## **Synchronization Manager Example**

MaxDB

© SAP AG 2005

## Contents

1	Introduction.....	3
1.1	Message Server .....	3
1.2	Synchronization Manager GUI .....	3
1.3	Synchronization Service.....	3
2	Example .....	4
2.1	The Database Instances .....	5
2.1.1	Predefined Names of the Example .....	5
2.2	The Tables .....	5
2.2.1	USER_MASTER.CUSTOMER .....	5
2.2.2	USER_MASTER.RESERVATION .....	6
2.2.3	USER1_CLIENT.CUSTOMER.....	6
2.2.4	USER2_CLIENT.GUEST .....	6
2.2.5	USER2_CLIENT.BOOKING .....	6
3	Setting up the Example .....	7
3.1	Prerequisites .....	7
3.2	Creating the User of the Synchronization Manager .....	7
3.3	Creating the Users and the Tables of the Example .....	7
3.4	Initializing the Message Server .....	9
3.5	Port Numbers Already in Use.....	9
4	Definition of the Synchronization Schema.....	10
4.1	Connection to the Master Database.....	10
4.2	Preparation of the Tables .....	10
4.2.1	Column Group.....	11
4.2.2	Definition of an Additional Column Group.....	11
4.3	Definition of the Master Unit 'CustomerUnit' .....	12
4.3.1	Add Table 'CUSTOMER' to 'CustomerUnit' .....	12
4.3.2	Add Client Unit to 'CustomerUnit' .....	13
4.4	Definition of the Master Unit 'ReservationUnit' .....	14
4.4.1	Add Tables to 'ReservationUnit'.....	14
4.4.2	Add Client Unit 'BookingUnit' to 'ReservationUnit'.....	15
4.5	Message Server Connection.....	18
4.6	Activation of the Units.....	18
4.7	Send Contents to Client Tables .....	19
4.7.1	Initialization of 'CustomerClientUnit' .....	19
4.7.2	Initialization of the Client Unit 'BookingUnit' .....	19
4.8	Quit the GUI Session.....	19
5	Synchronization Services .....	20
5.1	Synchronization Service at the Client Database .....	20
5.2	Synchronization Service at the Master Database .....	20
5.3	Terminating the Synchronization Services.....	20
6	Cleaning Up.....	20

# 1 Introduction

The synchronization manager handles a master database and a number of different client databases, where the client databases each have subsets of the data of the master database. All of the databases can concurrently access, update and modify data.

## 1.1 Message Server

The synchronization manager uses a message server to broadcast data changes from the master database to the client databases and to transport data change requests point-to-point from a client database to the master database in an orderly and asynchronous fashion.

The message server is implemented in accordance with the Java Message Service (JMS) specifications. In JMS relationships are defined as either “topics” (where messages associated with the “topic” are broadcast to all designated recipients of the topic) or “queues” (where messages are set through a queue in point-to-point fashion).

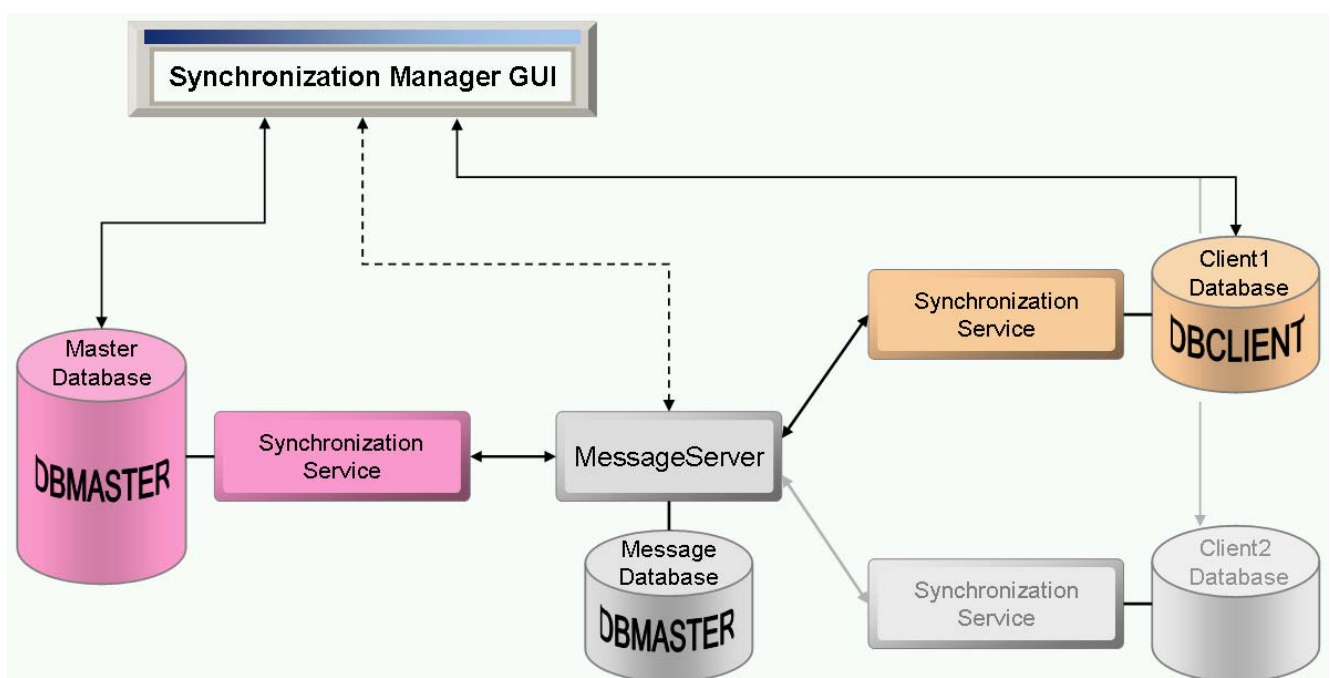
## 1.2 Synchronization Manager GUI

The synchronization manager GUI allows the service user of the master database to set up and manage the synchronization schema. The schema identifies the data to be shared asynchronously between the master database and the participating client databases. The service user groups the preexisting tables of the master database into replication units and defines the relationships to the preexisting empty tables of the client databases. A master-client relationship includes the mapping of the table and column names, the selection of columns, and the definition of column constraints.

The complete definition of the replication units and relations is stored at the master database. When a synchronization schema is created, each of the participating client databases will receive the relevant portions of the synchronization schema. Therefore the GUI establishes connections to the participating clients.

## 1.3 Synchronization Service

Each participating database is associated with a synchronization service. A synchronization service is a program coded to receive (and act upon) messages from the message server and to create and send messages to the message server. After the client databases has been initially synchronized with the master database, the services synchronize the concurrent data manipulation between the participating databases.



## 2 Example

The example is tested under Windows XP with Java 1.4.2 and J2EE 1.3.

The version of MaxDB has to be at least 7.6.00.11.

Two MaxDB database instances named DBMASTER and DBCLIENT are used for the synchronization example.

The tables of the message server are located at DBMASTER.

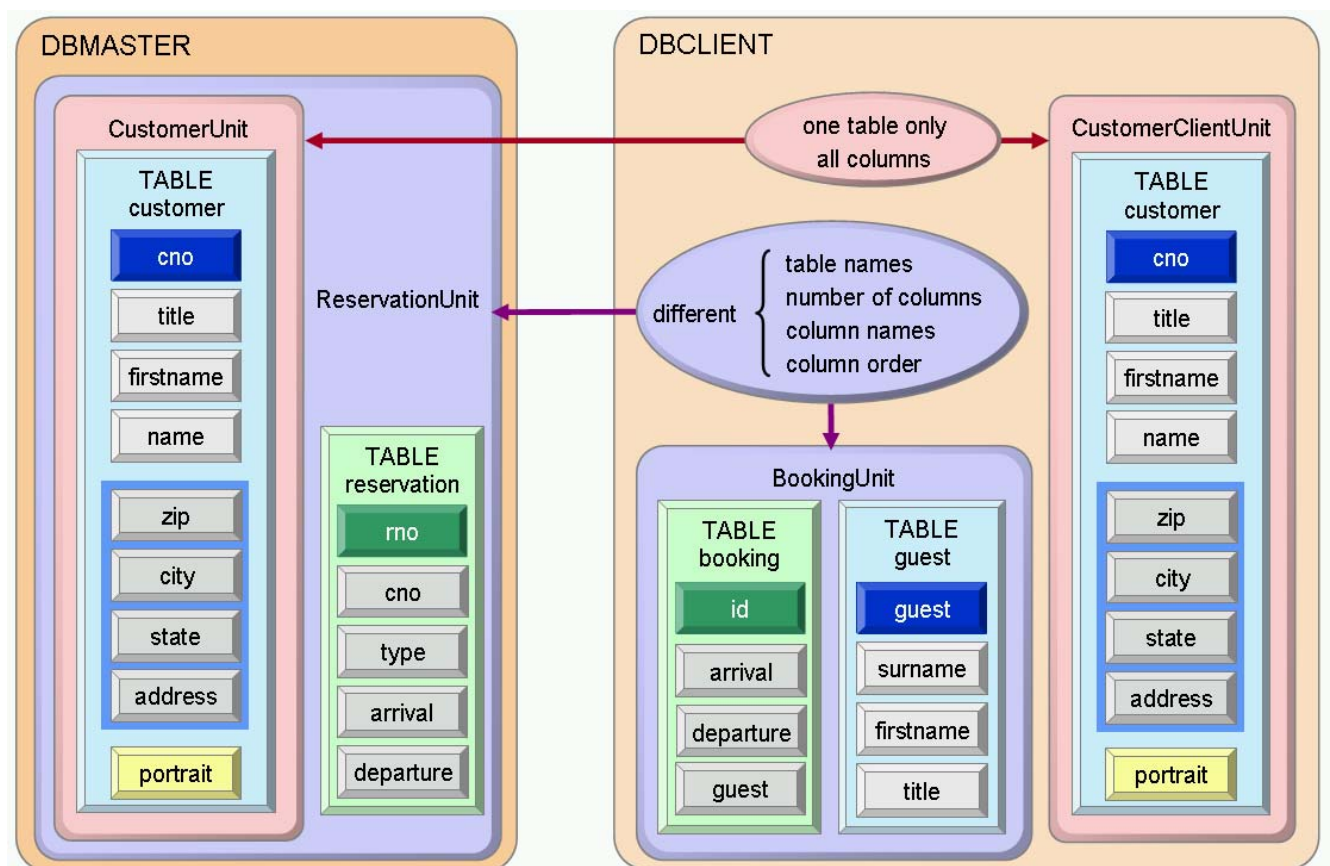
The master tables CUSTOMER and RESERVATION belonging to USER\_MASTER are located at DBMASTER.

The client table CUSTOMER belongs to USER1\_CLIENT and the client tables BOOKING and GUEST belong to USER2\_CLIENT.

Two replication units are defined:

- CustomerUnit  
Consists of the whole table CUSTOMER and is synchronized with CustomerClientUnit.
- ReservationUnit  
Consists of subsets of the tables CUSTOMER and RESERVATION and is synchronized with BookingUnit.

The synchronization between ReservationUnit and BookingUnit demonstrates the mapping of table and column names, the selection of columns, the handling of a different column order and the definition of a column constraint.



## 2.1 The Database Instances

### 2.1.1 Predefined Names of the Example

Database	URL
DBMASTER	jdbc:sapdb://127.0.0.1/DBMASTER
DBCLIENT	jdbc:sapdb://127.0.0.1/DBCLIENT

Database	User Type	User Name	Password	Owner
DBMASTER	operator	DBM	DBM	--
	administrator	DBADMIN	DBADMIN	--
	service user	DBSERVICE	SECRET	DBA
	example user	USER_MASTER	MASTER	DBSERVICE
DBCLIENT	operator	DBM	DBM	--
	administrator	DBADMIN	DBADMIN	--
	service user	DBSERVICE	SECRET	DBA
	example user	USER1_CLIENT	CLIENT	DBSERVICE
	example user	USER2_CLIENT	CLIENT	DBSERVICE

Database	Schema Type	Schema Name	Owner
DBMASTER	message server	MESSAGESERVICE	DBSERVICE
	synchronization service	SYNCHRONIZATIONSERVICE	DBSERVICE
DBCLIENT	synchronization service	SYNCHRONIZATIONSERVICE	DBSERVICE

Database	Example Tables	Owner / Schema Name
DBMASTER	CUSTOMER	USER_MASTER
	RESERVATION	USER_MASTER
DBCLIENT	CUSTOMER	USER1_CLIENT
	GUEST	USER2_CLIENT
	BOOKING	USER2_CLIENT

## 2.2 The Tables

### 2.2.1 USER\_MASTER.CUSTOMER

```
CREATE TABLE user_master.customer
(cno          INTEGER NOT NULL,
 title        CHAR (10),
 firstname    CHAR (20),
 name         CHAR (20),
 zip          CHAR (5),
 city         CHAR (20),
 state        CHAR (5),
 address      CHAR (30),
 portrait     LONG BYTE,
 PRIMARY KEY (cno))
```

cno	title	firstname	name	zip	city	state	address
3000	Mrs	Jenny	Porter	10580	New York	NY	1340 N.Ash Street, #3
3100	Mr	Peter	Brown	48226	Detroit	MI	1001 34th Str., APT.3
3200	Company	--	Datasoft	90018	Los Angeles	CA	486 Maple Str.
3300	Mrs	Rose	Brian	75243	Dallas	TX	500 Yellowstone Drive, #2
3400	Mrs	Mary	Griffith	20005	Washington	DC	3401 Elder Lane

cno	title	firstname	name	zip	city	state	address
3500	Mr	Martin	Randolph	60615	Chicago	IL	340 MAIN STREET, #7
3600	Mrs	Sally	Smith	75243	Dallas	TX	250 Curtis Street
3700	Mr	Mike	Jackson	45211	Cincinnati	OH	133 BROADWAY APT. 1
3800	Mrs	Rita	Doe	97213	Portland	OR	2000 Humboldt Str., #6
3900	Mr	George	Howe	75243	Dallas	TX	111 B Parkway, #23
4000	Mr	Frank	Miller	95054	Santa Clara	CA	27 5th Str., 76
4100	Mrs	Susan	Baker	90018	Los Angeles	CA	200 MAIN STREET, #94
4200	Mr	Joseph	Peters	92714	Irvine	CA	700 S. Ash Str., APT.12
4300	Company	-.-	TOOLware	20019	Washington	DC	410 Mariposa Str., #10
4400	Mr	Antony	Jenkins	20903	Silver Spring	MD	55 A Parkway, #15

### 2.2.2 USER\_MASTER.RESERVATION

```
CREATE TABLE user_master.reservation
(
  rno          INTEGER NOT NULL,
  cno          INTEGER,
  type         CHAR (10),
  arrival      DATE,
  departure    DATE,
  PRIMARY KEY (rno))
```

rno	cno	type	arrival	departure
100	3000	single	2004-11-13	2004-11-15
110	3000	double	2004-12-24	2005-01-06
120	3200	suite	2004-11-14	2004-11-18
130	3900	single	2005-02-01	2005-02-03
150	3600	double	2005-03-14	2005-03-24
140	4300	double	2004-04-12	2004-04-30
160	4100	single	2004-04-12	2004-04-15
170	4400	suite	2004-09-01	2004-09-03
180	3100	double	2004-12-23	2005-01-08
190	4300	double	2004-11-14	2004-11-17

### 2.2.3 USER1\_CLIENT.CUSTOMER

Table definition see USER\_MASTER.CUSTOMER.

### 2.2.4 USER2\_CLIENT.GUEST

```
CREATE TABLE user2_client.guest
(
  guest        INTEGER NOT NULL,
  surname      CHAR (20),
  firstname    CHAR (20),
  title        CHAR (10),
  PRIMARY KEY (guest))
```

### 2.2.5 USER2\_CLIENT.BOOKING

```
CREATE TABLE user2_client.booking
(
  id           INTEGER NOT NULL,
  arrival      DATE,
  departure    DATE,
  guest        INTEGER,
  PRIMARY KEY (id))
```

## 3 Setting up the Example

### 3.1 Prerequisites

- Your PATH environment variable must be set to include  
`<independent program path>/bin`  
and  
`<independent program path>/pgm.`
- If you are working under Unix/Linux, make sure that your account is in the `sdba` usergroup.
- A Java 1.4 runtime must be installed. You will be asked for its file system location by the configurator.
- Create and start two MaxDB instances named `DBMASTER` and `DBCLIENT` (as specified above) at the local computer.  
Use either the generic database tools (Windows: DbmGUI, Unix: dbmcli) or simply open a command prompt/shell, go to the `<independent program path>/app/syncman/example` directory, and run the program `createexemplatabases`.
- The `j2ee.jar` file from Sun must be provided by the user (since it can not be distributed with MaxDB). Make sure to know its location in the filesystem.
- You have to download the appropriate SWT runtime ( $\geq 3.01$ ) from [www.eclipse.org](http://www.eclipse.org) for your system. Place all the necessary jar files and shared libraries in the directory  
`<independent data path>/app/syncman/extern.`
- When any of the programs mentioned here is started for the first time, you will be asked for the locations of the java installation and the `j2ee.jar` file.
- **Reconfiguration**  
You can reconfigure the locations of the java installation and the `j2ee.jar` file at any time later by running the program `syncmanjconf`.

### 3.2 Creating the User of the Synchronization Manager

The user named `DBSERVICE` has to be created at the master and at the client database, these users can only be created by a user who has the `SYSDBA` privilege.

You can create the user with e.g. SQL Studio:

```
CREATE USER dbservice PASSWORD secret DBA NOT EXCLUSIVE REPLICATION
```

Or you can use the Database Manager GUI.

### 3.3 Creating the Users and the Tables of the Example

- Open a command prompt window (or a shell) and go to the directory `example`.
- Run the initialization program:  
`createexamplatables`

First the program drops

- the (possibly preexisting) users `USER_MASTER`, `USER1_CLIENT`, `USER2_CLIENT`
- the schemas `MESSAGESERVICE` and `SYNCHRONIZATIONSERVICE` of the user `DBSERVICE`.

The program creates the users and the example tables. The schemas are created when starting the message server and defining the synchronization schema.



### 3.4 Initializing the Message Server

- Start the message server using the program `msgserver`:

```
msgserver start
```

When `msgserver` is run the first time, it runs an interactive configuration wizard. Accept all the preset default values, except the database name, since for simplicity, this example will use the master database as message server database as well, so enter `DBMASTER` here:

<i>Database Host</i>	127.0.0.1
<i>Database Name</i>	DBMASTER
<i>Database User</i>	DBSERVICE
<i>Password of Database User</i>	SECRET
<i>Message Server Admin Password</i>	MSGSERVERADMIN
<i>Message Server Admin Port</i>	7220
<i>JNDI Port</i>	7221
<i>JMS Port</i>	7222

- The message server can be stopped by the command

```
msgserver stop
```

- Reconfiguration**

You can rerun the configuration wizard at any time later by giving the command

```
msgserver config
```

which gives you the previous values as defaults.

Don't stop the message server during the course of this example.

### 3.5 Port Numbers Already in Use

If a message server is already running or if one of the port numbers (7220, 7221, 7222) is already in use by another application, the following exception is displayed when starting the message server:

```
ServerException: Address already in use: JVM_Bind [Message Server, port 7222]
```

- The command

```
netstat -a
```

displays a list containing the used port numbers or service names of the current TCP/IP network connections:

TCP	P123456:7202	P123456...	LISTENING
TCP	P123456:xyz	P123456...	LISTENING
TCP	P123456:7222	P123456...	LISTENING

\ *port number*

- The port number of a service name (e.g. 'xyz') can be obtained from the file `WINDOWS\system32\drivers\etc\services`.
- Rerun the configuration wizard and set the affected ports to unused ones.
- Note that the `syncservice` configuration (below) refers to the ports you set here and specify them there accordingly. The `syncmangui` application must be started with explicit commandline arguments (see there) as well, if message server ports were modified.

## 4 Definition of the Synchronization Schema

- Open another command prompt window.
- Start the Synchronization Manager GUI:  
`syncmangui`
- If the message server runs with port numbers modified from the defaults, or on a different machine, explicit command line arguments must be provided:

```
syncmangui -msgserver_admin_port <port number>
           -msgserver_password <message server admin password>
           -msgserver_port <port number>
           -jndiserver_port <port number>
```

The arguments printed in bold are mandatory.

### 4.1 Connection to the Master Database

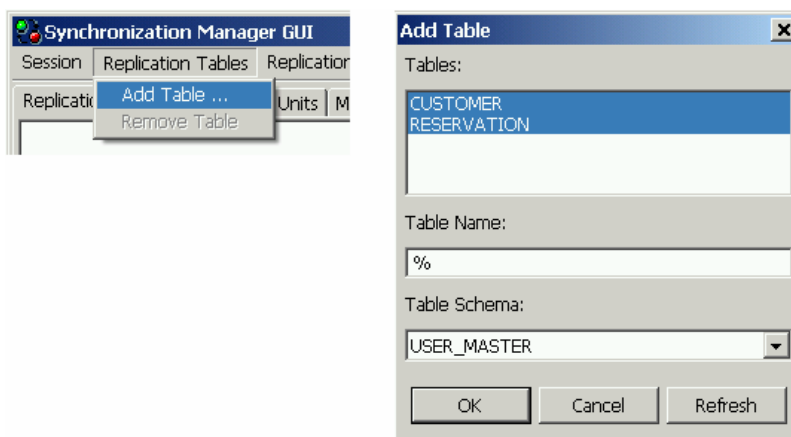
- Select *Session* → *Connect to Database*
- If the predefined names are used, type the following parameters to connect the service user to the master database:

<i>Host</i>	127.0.0.1
<i>Database</i>	DBMASTER
<i>User</i>	DBSERVICE
<i>Password</i>	SECRET

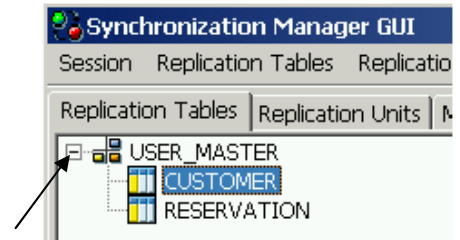
### 4.2 Preparation of the Tables

The preexisting tables of the master database can be selected for synchronization.

- Select *Replication Tables* → *Add Table*
- Choose the *Table Schema* USER\_MASTER
- Select the tables CUSTOMER and RESERVATION



- Click on the plus symbol to open the entry USER\_MASTER
- Mark the table CUSTOMER in order to display the table columns

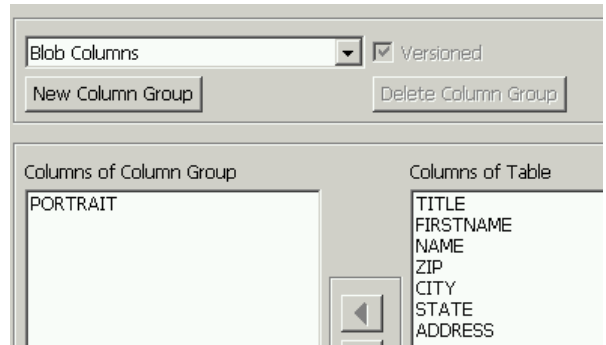


#### 4.2.1 Column Group

Since different clients can update different columns of the same row in parallel, the contents of a row may be inconsistent. E.g. one client updates the zip code while another client changes the name of the city. A column group combines columns to one logical unit in order to prevent the parallel modification of these columns.

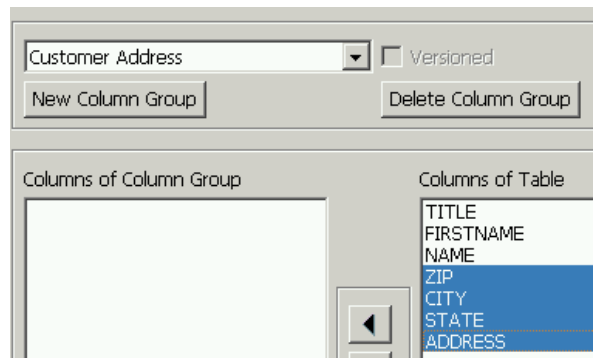
Once a client database updates a row of a synchronized table, a data change request message is sent to the master. This message includes the before images and/or the version of the affected columns and column groups. The before image and/or the version is matched against the current row in the master database to detect outdated change requests.

If a table contains blob columns, a versioned column group named 'Blob Columns' is automatically generated to avoid the data transfer of large before images. There can be only one versioned column group per table.



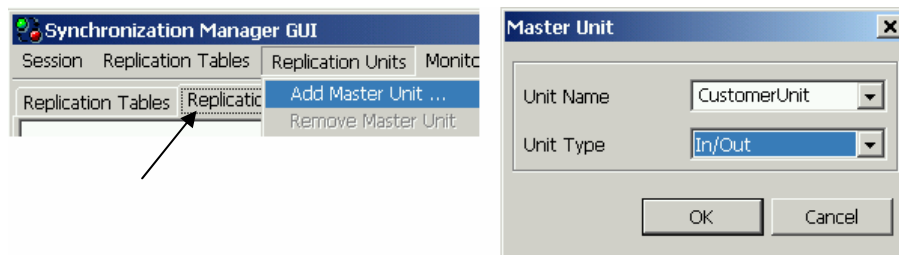
#### 4.2.2 Definition of an Additional Column Group

- Press the button *New Column Group* and define the group name  
Customer Address
- Mark the columns ZIP, CITY, STATE and ADDRESS
- Press the left arrow button



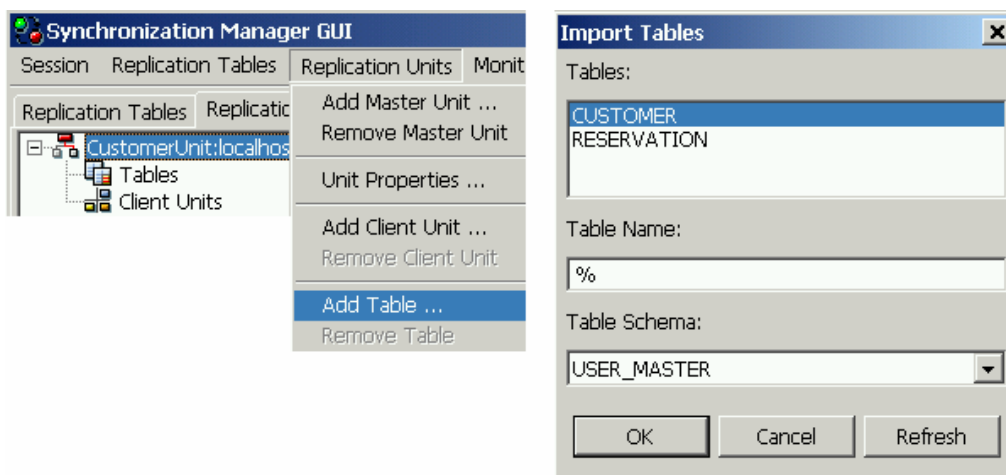
### 4.3 Definition of the Master Unit 'CustomerUnit'

- Switch to the tab *Replication Units*
- Select *Replication Units* → *Add Master Unit*
- Type the *Unit Name*  
CustomerUnit
- Select the *Unit Type* In/Out



#### 4.3.1 Add Table 'CUSTOMER' to 'CustomerUnit'

- Mark the entry CustomerUnit
- Select *Replication Units* → *Add Table*
- Import the table CUSTOMER



### 4.3.2 Add Client Unit to 'CustomerUnit'

- Select *Replication Units* → *Add Client Unit*
- Type the *Unit Name*  
CustomerClientUnit
- Enter the connection parameters of the client database

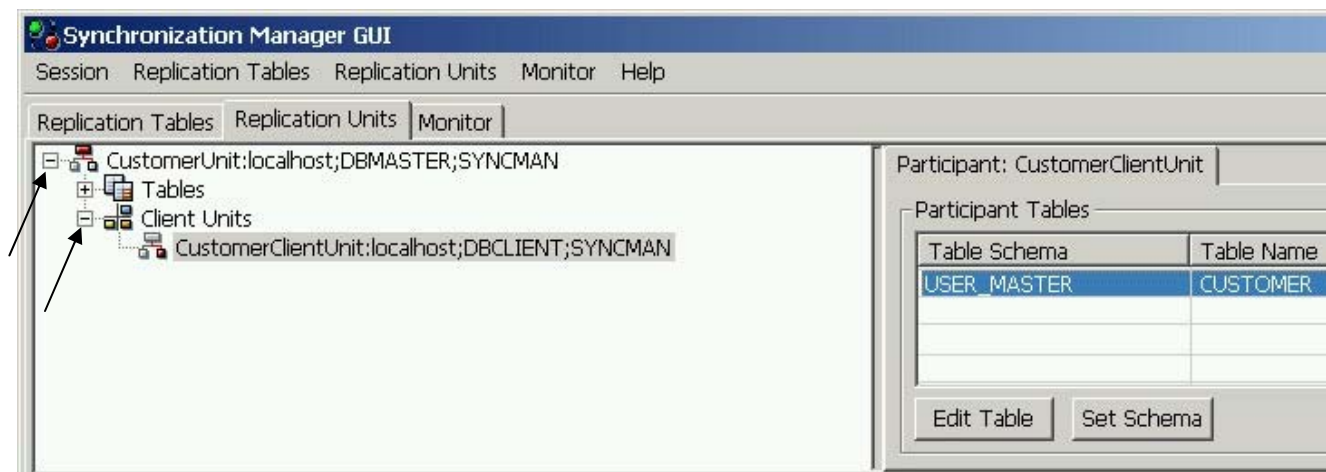
Predefined values:

*Host*        127.0.0.1  
*Database*   DBCLIENT  
*User*        DBSERVICE  
*Password*   SECRET

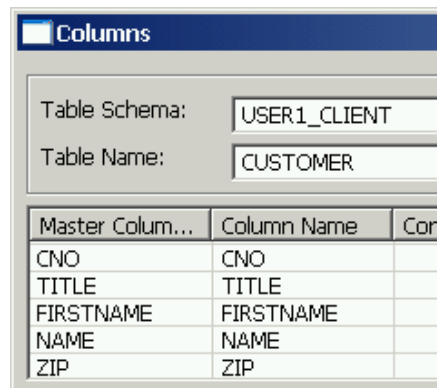
- *Save Password* may be selected
- Select the *Unit Type* 'In/Out'

#### 4.3.2.1 Table Name Mapping

- Click on the plus symbol to open the entry CustomerUnit
- Click on the plus symbol of the entry *Client Units*
- Mark the entry CustomerClientUnit in order to display the associated table
- Mark the table CUSTOMER
- Press the button *Edit Table*



- Change the *Table Schema*:  
USER\_MASTER → USER1\_CLIENT
- Press the *OK* button

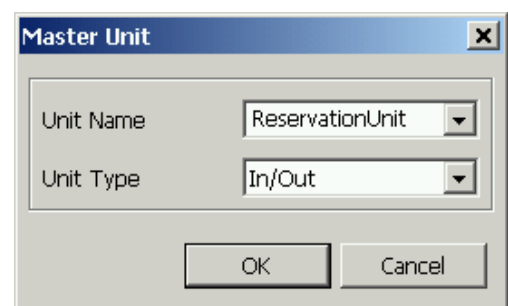


The 'Columns' dialog box shows the configuration for a table schema. The 'Table Schema' is set to 'USER1\_CLIENT' and the 'Table Name' is 'CUSTOMER'. Below, a table lists columns to be synchronized:

Master Column...	Column Name	Con:
CNO	CNO	
TITLE	TITLE	
FIRSTNAME	FIRSTNAME	
NAME	NAME	
ZIP	ZIP	

#### 4.4 Definition of the Master Unit 'ReservationUnit'

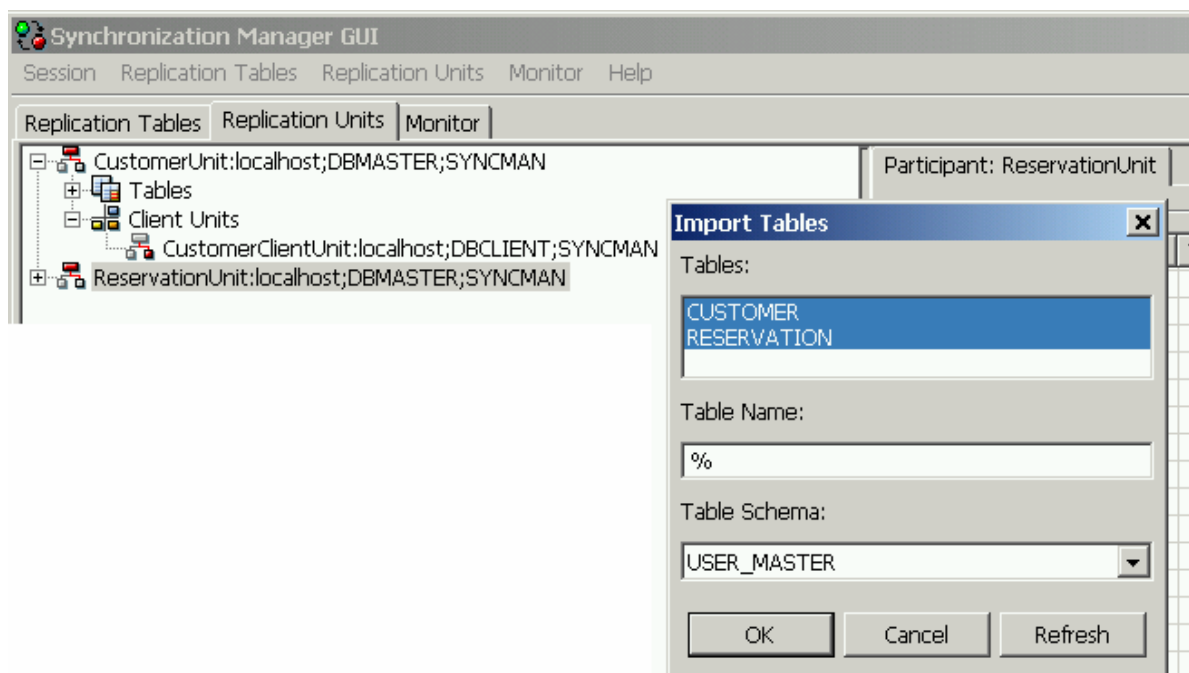
- Select *Replication Units* → *Add Master Unit*
- Type the *Unit Name*  
ReservationUnit
- Select the *Unit Type* 'In/Out'



The 'Master Unit' dialog box shows the configuration for a new master unit. The 'Unit Name' is set to 'ReservationUnit' and the 'Unit Type' is set to 'In/Out'. The 'OK' and 'Cancel' buttons are at the bottom.

##### 4.4.1 Add Tables to 'ReservationUnit'

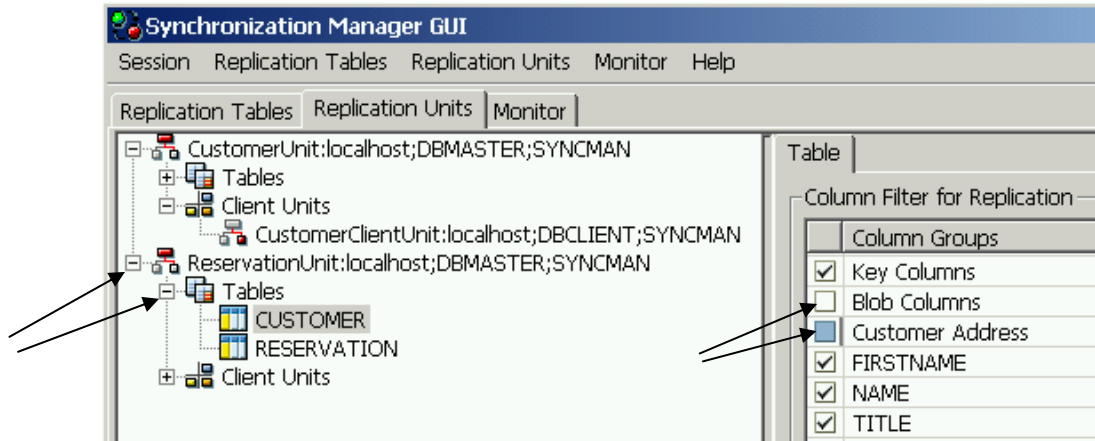
- Mark the entry *ReservationUnit*
- Select *Replication Units* → *Add Table*
- Import the tables *CUSTOMER* and *RESERVATION*



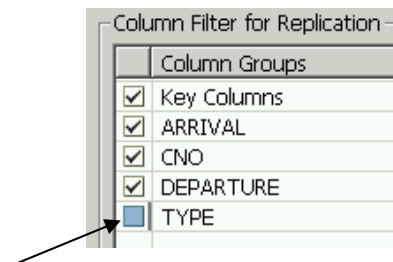
The screenshot shows the 'Synchronization Manager GUI' with the 'Replication Units' tab selected. The 'ReservationUnit:localhost;DBMASTER;SYNCMAN' entry is highlighted. An 'Import Tables' dialog box is open, showing a list of tables to be imported: 'CUSTOMER' and 'RESERVATION'. The 'Table Name' field is set to '%', and the 'Table Schema' is set to 'USER\_MASTER'. The 'OK', 'Cancel', and 'Refresh' buttons are at the bottom.

#### 4.4.1.1 Column Selection

- Click on the plus symbol to open the entry *ReservationUnit*
- Click on the plus symbol of the entry *Tables*
- Mark the table *Customer* in order to display the column groups
- Deselect the column groups 'Blob Columns' and 'Customer Address'



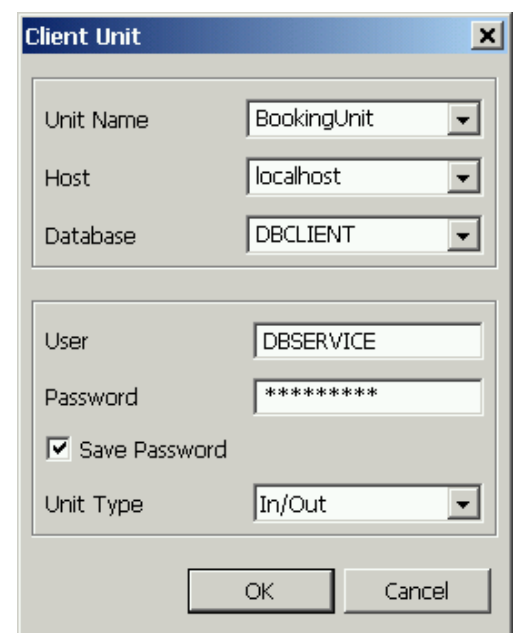
- Mark the table *Reservation*
- Deselect the column 'TYPE'



#### 4.4.2 Add Client Unit 'BookingUnit' to 'ReservationUnit'

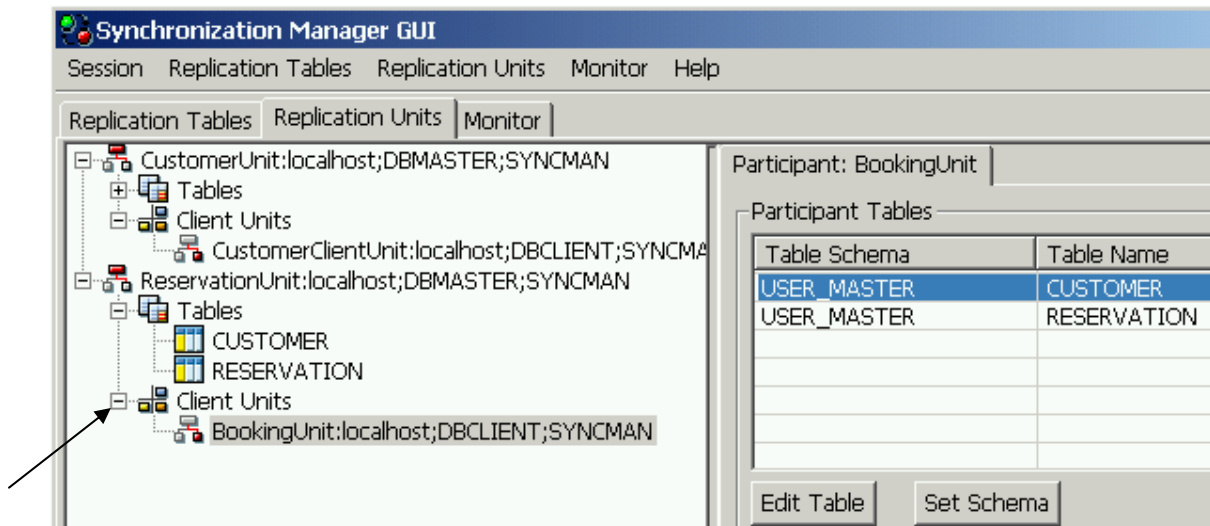
- Mark the entry *ReservationUnit*
- Select *Replication Units* → *Add Client Unit*
- Type the *Unit Name*  
BookingUnit
- Enter the connection parameters of the client database  
Predefined values:  

<i>Host</i>	127.0.0.1
<i>Database</i>	DBCLIENT
<i>User</i>	DBSERVICE
<i>Password</i>	SECRET
- *Save Password* may be selected
- Select the *Unit Type* 'In/Out'

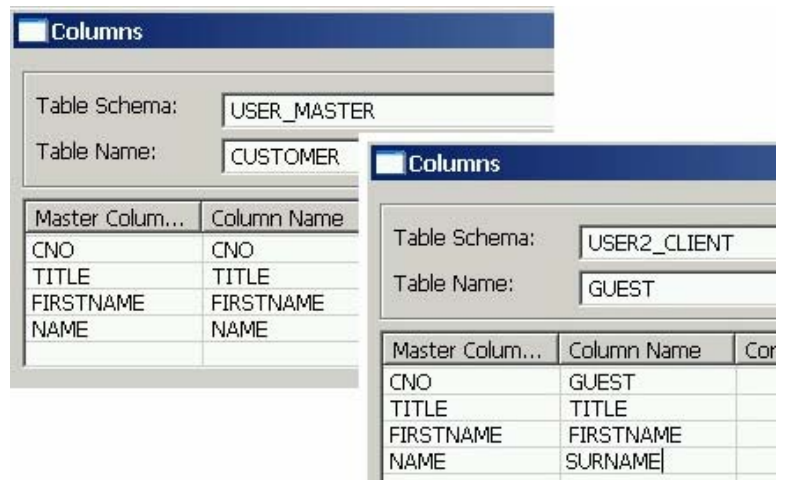


#### 4.4.2.1 Table and Column Name Mapping

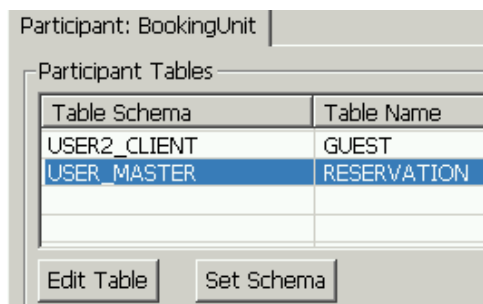
- Click on the plus symbol of the entry *Client Units* belonging to *ReservationUnit*
- Mark the entry *BookingUnit* in order to display the associated tables
- Mark the table *CUSTOMER*
- Press the button *Edit Table*



- Perform the following changes:
  - USER\_MASTER → USER2\_CLIENT
  - CUSTOMER → GUEST
  - CNO → GUEST
  - NAME → SURNAME
- Press the *OK* button



- Mark the table *RESERVATION*
- Press the button *Edit Table*





- Perform the following changes:
  - USER\_MASTER → USER2\_CLIENT
  - RESERVATION → BOOKING
  - RNO → ID
  - CNO → GUEST

Columns		
Table Schema:	USER2_CLIENT	
Table Name:	BOOKING	
Master Colum...	Column Name	C
RNO	ID	
CNO	GUEST	
ARRIVAL	ARRIVAL	
DEPARTURE	DEPARTURE	

#### 4.4.2.2 Column Constraint

- Add the *Constraint Type*
  - >=
 to the column ARRIVAL
- Add the *Constraint Value1*
  - 2004-11-01
 to the column ARRIVAL
- Press the *OK* button

Master Colum...	Column Name	Constraint Type	Constraint Value1	Constraint Value2	SQL Type Na...
RNO	ID				INTEGER
CNO	GUEST				INTEGER
ARRIVAL	ARRIVAL	>=	2004-11-01		DATE
DEPARTURE	DEPARTURE				DATE

## 4.5 Message Server Connection

The activation of the replication units requires a connection to the message server to create the topics (broadcast connections) and queues (point-to-point connections).

- Select *Session* → *Connect to Message Server*
- If the predefined values are used, type the following parameters:

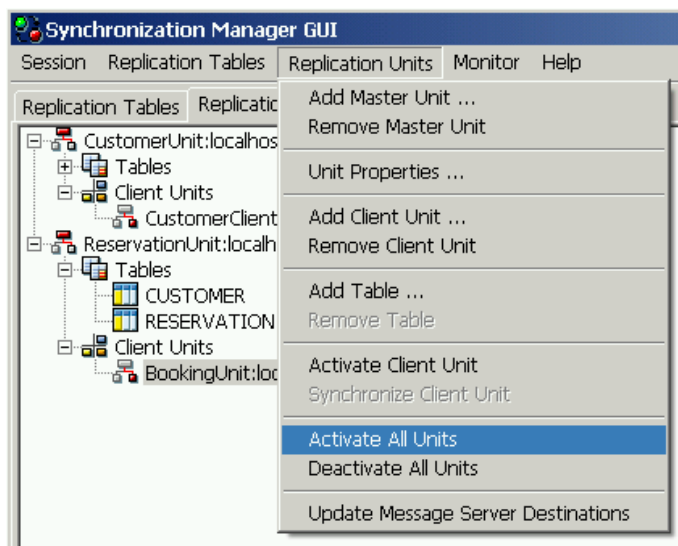
<i>Host</i>	127.0.0.1
<i>Port</i>	7220
<i>Password</i>	MSGSERVERADMIN

## 4.6 Activation of the Units

During the activation all necessary triggers, shadow tables and version tables are created in the involved database instances.

- Select *Replication Units* → *Activate All Units*

Deactivated Units are marked with red icons. Blue icons indicate activated units.



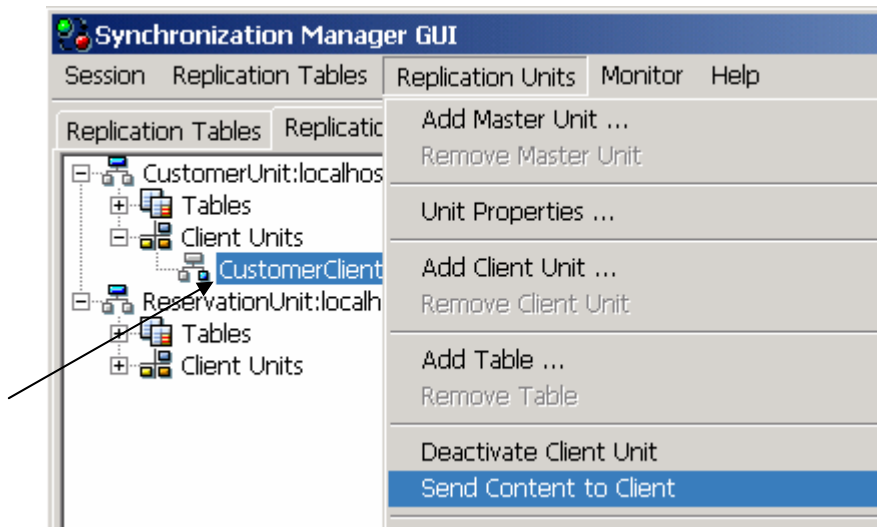
## 4.7 Send Contents to Client Tables

Before starting the synchronization services the first time, the involved tables at the client databases have to be empty. The synchronization of a client unit fills the involved client tables and initializes internal row version tables of the master and the client database.

### 4.7.1 Initialization of 'CustomerClientUnit'

The contents of the master table `CUSTOMER` is sent to the message server in order to initialize the client table `USER1_CLIENT.CUSTOMER`.

- Mark the entry `CustomerClientUnit`
- Select *ReplicationUnits* → *Send Content to Client*



After successful synchronization the following message is displayed in the status bar:

Initial Content Distribution 'CustomerClientUnit': 15 messages sent

### 4.7.2 Initialization of the Client Unit 'BookingUnit'

The contents of a subset of the master tables `CUSTOMER` and `RESERVATION` is sent to the message server in order to initialize the client tables `USER2_CLIENT.GUEST` and `USER2_CLIENT.BOOKING`.

- Mark the entry `BookingUnit`
- Select *ReplicationUnits* → *Send Content to Client*

After successful synchronization the following message is displayed in the status bar:

Initial Content Distribution 'BookingUnit': 22 messages sent

## 4.8 Quit the GUI Session

- Select *Session* → *Disconnect Database*
- Select *Session* → *Disconnect Message Server*
- Select *Session* → *Exit*

## 5 Synchronization Services

### 5.1 Synchronization Service at the Client Database

- To start the Synchronization Service at the client database, type in a command prompt window  
`syncservice start -d DBCLIENT`
- When `syncservice` is started the first time for a specific database instance, it runs an interactive configuration wizard. Accept all the preset default values (except possibly altered messageserver ports). Since the message server runs at the local machine in this example, enter 127.0.0.1 in the 'message server host' field:

<i>Synchronization User</i>	DBSERVICE
<i>Synchronization User Password</i>	SECRET
<i>Message Server Host</i>	127.0.0.1
<i>Synchronization Service Admin Port</i>	7223
<i>JNDI Port of Message Server</i>	7221
<i>Service Port of Message Server</i>	7222

- **Reconfiguration**

You can reconfigure the Synchronization Service for a specific database instance at any time later by giving the command

```
syncservice config -d <database instance>
```

First the synchronization service receives the messages of the initial synchronization and fills the client tables.

### 5.2 Synchronization Service at the Master Database

- Open an additional command prompt window.
- Start the Synchronization Service of the master:  
`syncservice start -d DBMASTER`
- In the configuration wizard, change the default of 7223 for the SyncService admin port to (e.g.) 7224, since this is the second SyncService running on the local machine.

Now the synchronization services of the master and client database are running and data changes of the involved tables are replicated bidirectionally.

The following commands stop the synchronization services of the client and the master:

```
syncservice stop -d DBCLIENT  
syncservice stop -d DBMASTER
```

## 6 Cleaning Up

Go to the directory '`syncman\example`' and call the program `createexamplatables` (see above) with the option '`-clear`' in order to drop the database schemas `MESSAGESERVICE` and `SYNCHRONIZATIONSERVICE` and to drop the users of the example (`USER_MASTER`, `USER1_CLIENT` and `USER2_CLIENT`).

You can drop the example databases by calling '`createexempldatabases -drop`'.