15. Customising BBMS

What is BBMS?

BBMS stands for Browser Based Maintenance System. It is the administrative interface used to maintain your store and to produce real time reports.

From a technical perspective BBMS is simply another user group. As such there is nothing within BBMS that cannot be achieved in any other user group. BBMS is simply a selection of pages using iNETstore tags and commands and SQL statements.

Customising these pages is no different to customising any other page within any other user group. Customising the features and functions within the pages requires a knowledge of the iNETstore tag and command language. A knowledge of SQL is also an advantage. These concepts are explained in greater detail in the Appendix.

Adding fields to BBMS

If you have added extra fields to the database you will probably want to add them to BBMS so that the information contained within these fields can be maintained along with the rest of the store data.

Adding extra fields to BBMS is not complex and can be achieved without a great understanding of the iNETstore tag and command language.

To illustrate adding extra fields to BBMS we will use a hypothetical example of a book store. If you were running an online book store you may want to include item fields such as:

- Author
- ISBN
- Publisher
- Binding
- Format
- Review

These are item fields and as such would normally be included in BBMS on the page edit-itm.ehtml.

Most pages within BBMS begin with a SQL statement that displays the existing content for the record being displayed. The SQL statement in edit-itm.ehtml begins:

```
[DBxBEGIN_SQL "
SELECT *
FROM ITM[DBxDBNAME]
WHERE id=[DBxARGS_id]"]
```

The key is to ensure that the SQL statement calls the fields that you wish to add. In the statement above all fields are called by virtue of the asterisk (i.e. SELECT *).

Then all that is required is to display the field that you wish to call. As we learn in our SQL syntax (explained further in the Appendix) the format of the tag is:

[DBxSQL_fieldname]

where fieldname = the name of the field that you wish to display from within the SQL statement

Therefore in our example, the tags would be:

[DBxSQL_Author]
[DBxSQL_ISBN]
[DBxSQL_Publisher]
[DBxSQL_Binding]
[DBxSQL_Format]
[DBxSQL_Review]

The final task is to map the name of an input field to that of the field. The submit will fail unless the name of the inputs are correctly mapped to a corresponding fieldname in the database.

So to complete our inputs for our bookstore we would the add the following inputs to edit-itm.ehtml:

```
<input type="text" name="author" value="[DBxSQL_author]">
<input type="text" name="isbn" value="[DBxSQL_ISBN]">
<input type="text" name="publisher" value="[DBxSQL_publisher]">
<input type="text" name="binding" value="[DBxSQL_binding]">
<input type="text" name="format" value="[DBxSQL_format]">
<input type="text" name="review" value="[DBxSQL_review]">
```

Note how the “name=” for each input matches the field name. The “value=” will display the current value in the database for the record being displayed.