



# DotCom HRS Sooplet Tutorial

---

## Introduction

The following lessons provide an introduction to using **Instant Soop Version 4.0** or later, guiding you through the stages of creating a Human Resource Management **Sooplet** for DotCom Industries – ‘DotCom HRS’. This will enable DotCom to manage information about their employees and organisational structure, answer enquiries and produce corporate correspondence.

Most lessons only take a minute or two; the full tutorial requires between 1 and 2 hours to complete. By the end of the tutorial you will have built and deployed a working HRS Solution for DotCom Industries.

You are advised to tackle the lessons in order to ensure your Sooplet design stays in step, but you can exit the application between lessons.

As the lessons require you to switch between the Instant Soop Designer, and your DotCom Sooplet, the instructions have been colour coded – **blue for the Designer** and **red for the Sooplet** to assist you.

The text that you need to type is printed in **this bold font**.

Actions that you need to perform are numbered 1, 2, 3, etc.

Menu options and button names are printed in italics e.g. *Add Class*.

Things to note are highlighted with the following symbol.



Things that might go wrong are highlighted with the following symbol.



Design issues are highlighted with a light bulb.



The end of a lesson is marked with the following symbol.



*Good luck!*

---

## Table of Contents

---

Lesson 1 – Create a new Sooplet and Class .....	1
Lesson 2 – Create Auto-Text Fields and Employee Objects.....	2
Lesson 3 – Create a Mandatory Date Field .....	5
Lesson 4 – Initializers and Prefixes .....	6
Lesson 5 – Badge Settings.....	8
Lesson 6 – Create a Mandatory Relationship.....	9
Lesson 7 – Editing Help .....	11
Lesson 8 – Creating a Self-Relationship .....	13
Lesson 9 – Creating a Calculated Field.....	15
Lesson 10 – Creating a Method .....	16
Lesson 11 – Securing the Sooplet.....	19
Lesson 12 – Placing Shortcuts.....	23
Lesson 13 – Inheritance.....	24
Lesson 14 – Querying Data .....	27
Lesson 15 – Generating Reports.....	28
Lesson 16 – Creating a Document .....	29
Lesson 17 – Saving Sooplet as Script .....	31
Lesson 18 –Deploying a Sooplet to SoopletServer .....	32
Lesson 19 –Distribute a Sooplet via the Web .....	34

---

## Lesson 1 – Create a new Sooplet and Class

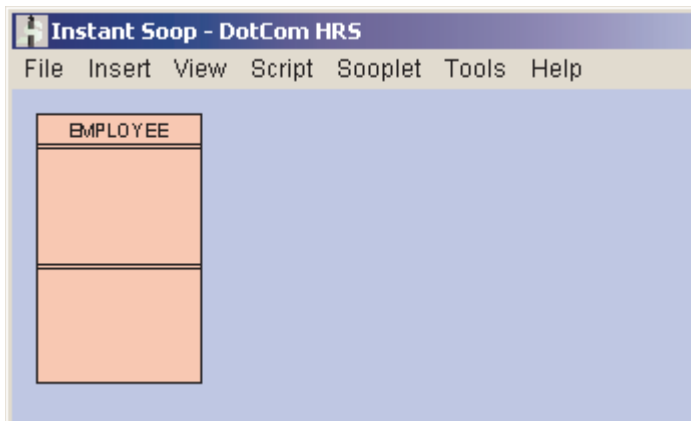
---

1. Open up the Instant Soop designer application and choose the *New* option from the *File* menu.
2. Enter **DotCom HRS** in the 'Sooplet Name' Box and click *O.K.*

You will be presented with a blank design diagram on which to layout your Sooplet. Minimise the smaller Sooplet window that appears, if necessary.

3. Right-click the diagram and choose *Add Class* from the pop-up menu.
4. Our first Class is going to be the EMPLOYEE Class, so enter **EMPLOYEE** as the Class Name, and click *O.K.*

The EMPLOYEE Class will appear in your diagram, as illustrated.



Lesson 1 is complete. You have created the Sooplet **DotCom HRS**, and your first Class, **EMPLOYEE**. It doesn't do much yet, but in the next lesson we will add some fields and create some Employee objects.

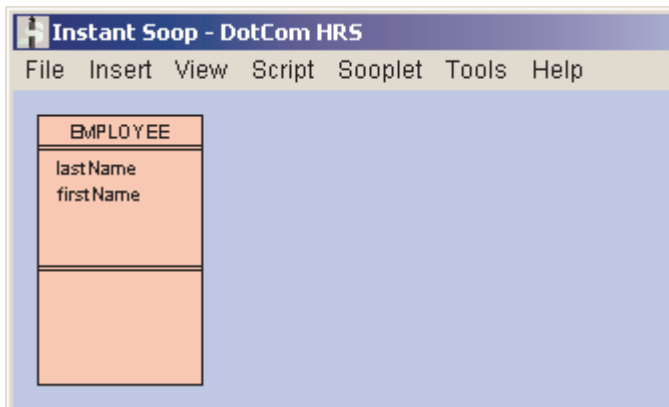
## Lesson 2 – Add Fields and create Employee Objects

---

We now need to add some Fields to our EMPLOYEE Class, to hold data about each employee.

1. Right-click the EMPLOYEE Class on the diagram and choose *Add Field* from the Class pop-up menu. The 'Add New Field' dialog will be displayed.
2. Enter **lastName** in the Field Name box; ensuring the mix of upper and lower case characters is as shown. Click *O.K.* to add the field.
3. Repeat the above steps to add the **firstName** field.

Your diagram should now look like this...

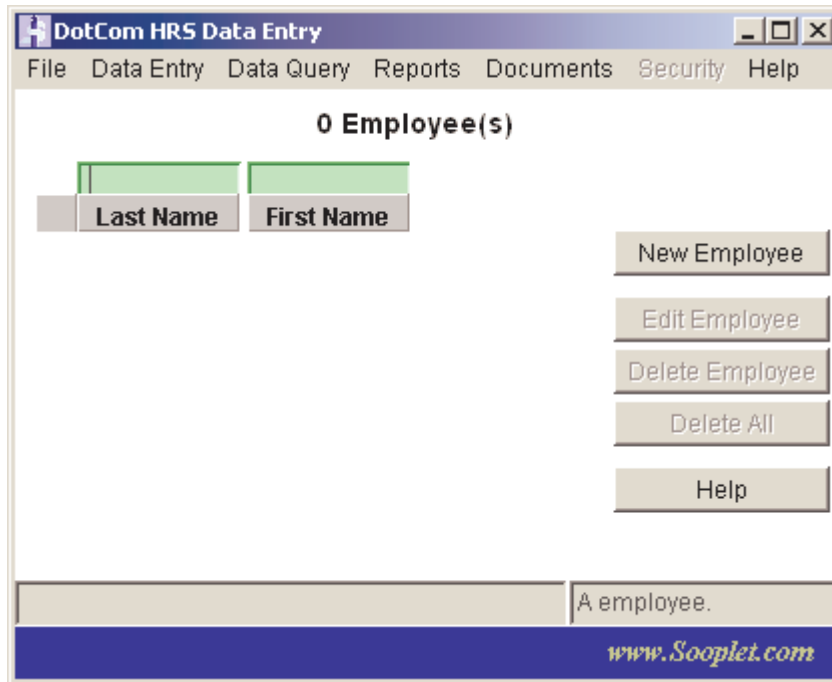


and you have a functional design!



4. Maximise the DotCom HRS window on your taskbar (or choose *Preview Sooplet* from the *Sooplet* menu if this window has closed).

Your Sooplet should look similar to this:



Please note a number of features about your Sooplet:

- The Sooplet name appears in the Window's title bar
- The Employee collection appears in the *Data Entry* and *Data Query* menus
- The lastName and firstName fields have appeared as column headings in the table, in presentation format.
- Help for the EMPLOYEE Class is automatically available by clicking the *Help* Button. In Lesson 7 we will tailor the help text to our particular needs.

Employee 'Objects' can now be added to our DotCom HRS Sooplet.

5. In *Data Entry View*, which is the default on opening a Sooplet, click the *New Employee* Button and the following *Data Entry Form* will be displayed.

6. Enter names for Last Name and First Name, and then click *O.K.* to create the first Employee Object.
7. Repeat the previous two steps to add 3 more employees of DotCom Industries.



Note how the field formats have been automatically set to text, and drop-down lists are provided on the **Data Entry Form** to assist entry of recurring data.

Your Sooplet Data entry View of Employees should now look like the following:

Last Name	First Name
White	Mary
Williams	Sarah
Jones	Bill
▶ Smith	Fred



Congratulations, Lesson 2 is complete! You have added some auto-text fields to your EMPLOYEE Class, and created some Employee Objects. You now have a working application. In Lesson 3 we will add a field with a fixed data type.

## Lesson 3 – Create a Mandatory Date Field

---

The Fields `lastName` and `firstName` have been created as **automatic format** Fields by default. Automatic format Fields learn about the type of data as it is entered. This is particularly useful for rapid prototyping, but may result in incorrect data being input. We are now going to create a new Field for the employee's start date, which has a **fixed format**. We will also make the Field **mandatory**.

1. In the Instant Soop Designer window, add a new Field `startDate` to the EMPLOYEE Class, as you did in the previous lesson.
2. Right-click the `startDate` Field, to display the Field pop-up menu, and choose *Set Data Type*. The 'Set Data Type' dialog will be displayed.
3. Clear the *Auto* checkbox, and select the *Date* Option.
4. Select the *Mandatory* option and click *O.K.*
5. In the DotCom Sooplet Data Entry View of Employees, select an Employee Object by clicking a row in the table, and click the *Edit Employee* button. The **Data Entry Form** will open.
6. Click the widget button [...] adjacent to the Start Date Field, to display the Calendar Widget.



Note how the Calendar Widget is provided the first time you edit an Employee Object, because only dates can be entered in this fixed format field:

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Also note you cannot close the form, saving changes with O.K., without entering a value for the Start Date from the Calendar. You can always exit a form without saving changes by clicking the *Cancel* button.

7. Complete this lesson by setting the start dates for all employees, ensuring that at least one is in the future – for an employee who has yet to commence employment with DotCom.



Lesson 3 is complete. In the next lesson we will extend the design by adding a second Class to our Sooplet.

## Lesson 4 – Initializers and Prefixes

---

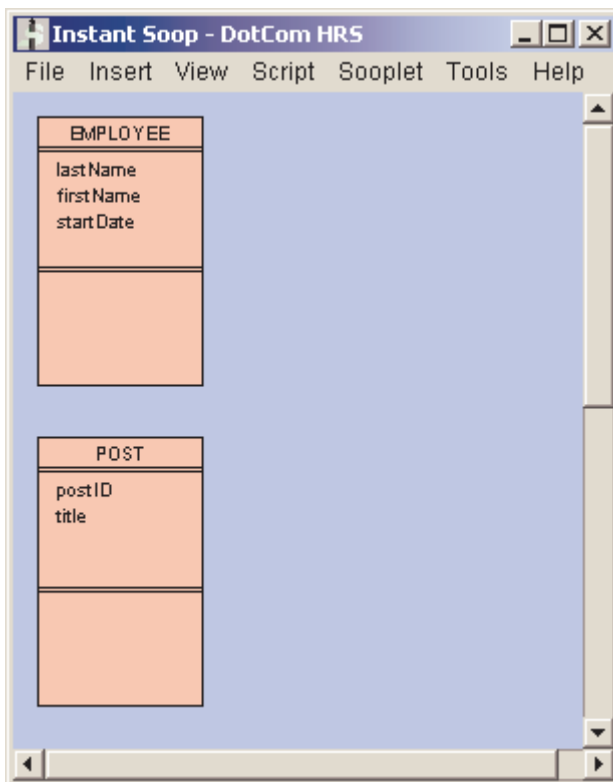
Having created the EMPLOYEE Class, we now require a Class to represent the Employee's job, the POST Class.



You will see in later lessons why the employee's job is best represented by its own Class, rather than a field in the EMPLOYEE Class.

1. Right-click the diagram and choose *Add Class*.
2. Enter **POST** as the Class Name in the 'Add New Class' dialog and then click *O.K.*
3. Add Fields to the POST Class for **postID** and **title**

Your diagram should now look like the following:



4. We require the postID to be automatically generated for each Post Object. This requires a Field Initializer. Right-click the postID Field and choose *Set Initializer* from the Field menu.
5. Select the *AutoNumber* option and click *O.K.*

6. As DotCom Industries is growing rapidly, we require a prefix to all postIDs to distinguish them from posts inherited from acquisitions. Right-click the postID Field and choose *Edit Properties* from the Field menu.
7. Enter **DC** as the Prefix for this field, and click *O.K.*
8. In the DotCom Sooplet, select **POST** from the *Data Entry* menu.
9. Create exactly 3 new Post Objects, entering job titles of your choosing e.g. CEO, Sales Manager and Sales Person.



Note how the postID is automatically incremented, prefixed by DC and made Read-Only.



You have completed Lesson 4, illustrating the use of Initializers and Field Properties. In lesson 6 we will link the EMPLOYEE Class to the POST Class, to enable employees to have jobs, but first a short lesson on Badges.

## Lesson 5 – Badge Settings

---

A Badge is a subset of data that describes a specific object in a recognizable way.



New Classes are created with a default Badge Setting of 1, i.e. the first field is used for the Badge. There could be many employees with the same surname, and throughout our Sooplet we need to be able to accurately select a particular Employee Object. So `lastName` is not sufficient to identify an employee, but `lastName` and `firstName` should be.

1. In the diagram, right-click the **EMPLOYEE Class Heading** to display the **Class menu**.
2. Choose *Set Badge* from the menu, and the ‘Set Badge’ dialog will be displayed. For Employees, we need to increase the setting to [`lastName firstName`], so click the + button and then click *O.K.*
3. Repeat this procedure for the **POST Class**, so its Badge includes `postID` and `title`.



Re-open the **Data Entry View of Employees** from the menu, and note how **Badge field columns** group together on the left of the table. When the Class has more fields, the **Badge fields** will not scroll horizontally when other fields do.



This completes Lesson 5. In the next lesson on relationships, you will see other effects of the changes made to **Badge Settings**.

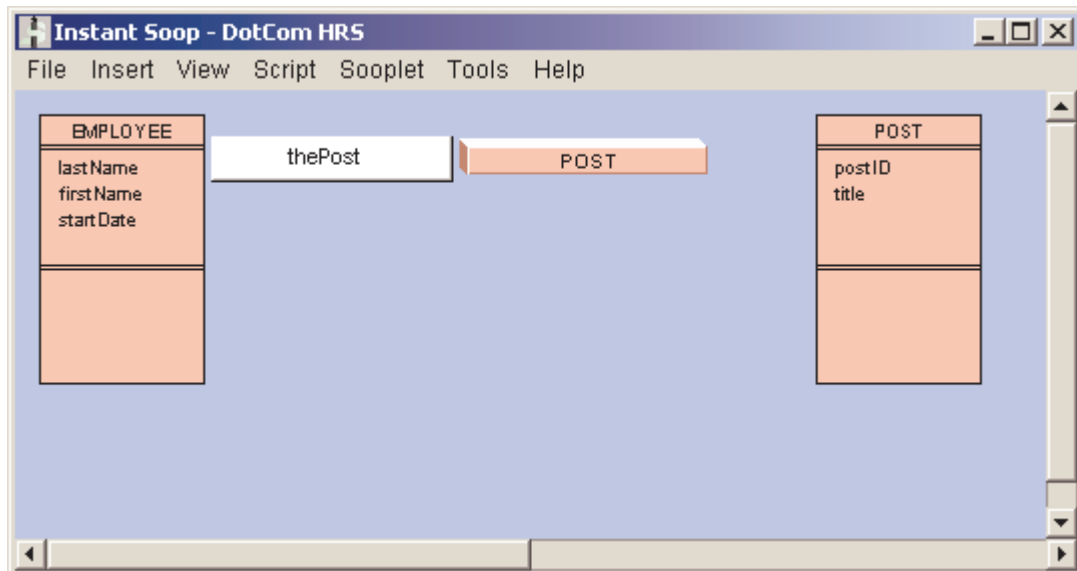
## Lesson 6 – Create a Mandatory Relationship

---

Every employee at DotCom Industries must occupy a single post. To implement this we need to link the EMPLOYEE Class to the POST Class with a One-To-One Relationship.

1. In the diagram, drag the POST Class over the EMPLOYEE Class. This is done by left-clicking the POST Class Heading and, keeping the mouse button pressed, dragging the Arrowed cursor over the EMPLOYEE Class before releasing the mouse button.
2. The 'Create Relationship' dialog will be displayed, with the EMPLOYEE and POST Classes selected as Source and Target Classes. One-To-One is the default mode for Relationships, so this does not need changing, and we can accept the default name for the Relationship, **thePost**. Select the *Mandatory* option before clicking *O.K.* to create the Relationship. Click *O.K.* to dismiss the dialog warning about Reverse Relationships, which we will consider later.

Your diagram should look like this:



We now need to associate each Employee Object with a Post Object.

3. In the DotCom Sooplet, select *EMPLOYEE* from the *Data Entry* menu. One-to-One associations are established in the Data Entry Form. Double-click each of the first 3 Employee rows, which is a quick way to edit employee details. Allocate a different post to each employee. In addition, try to *O.K.* the form details without allocating a post.



Note how the Posts are presented as a set of option buttons. Also note how the Mandatory condition was enforced, by displaying a warning and preventing exit from the form.

4. Add one more Post Object representing a direct report of an existing post, using the *Data Entry* menu, and re-edit the final Employee to fill this post.



Note how the Posts are now provided in a drop-down list as opposed to a set of option buttons, due to the number of choices available.



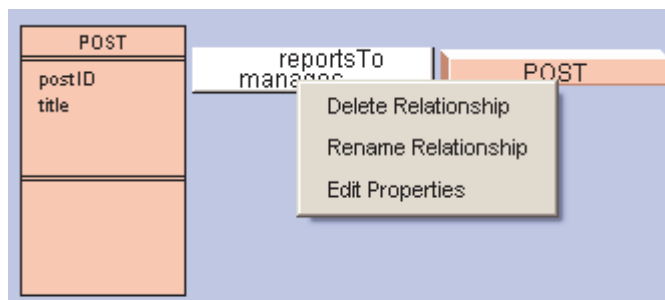
This completes Lesson 6. In Lesson 7 we will look at customising the Help facilities.

## Lesson 7 – Editing Help

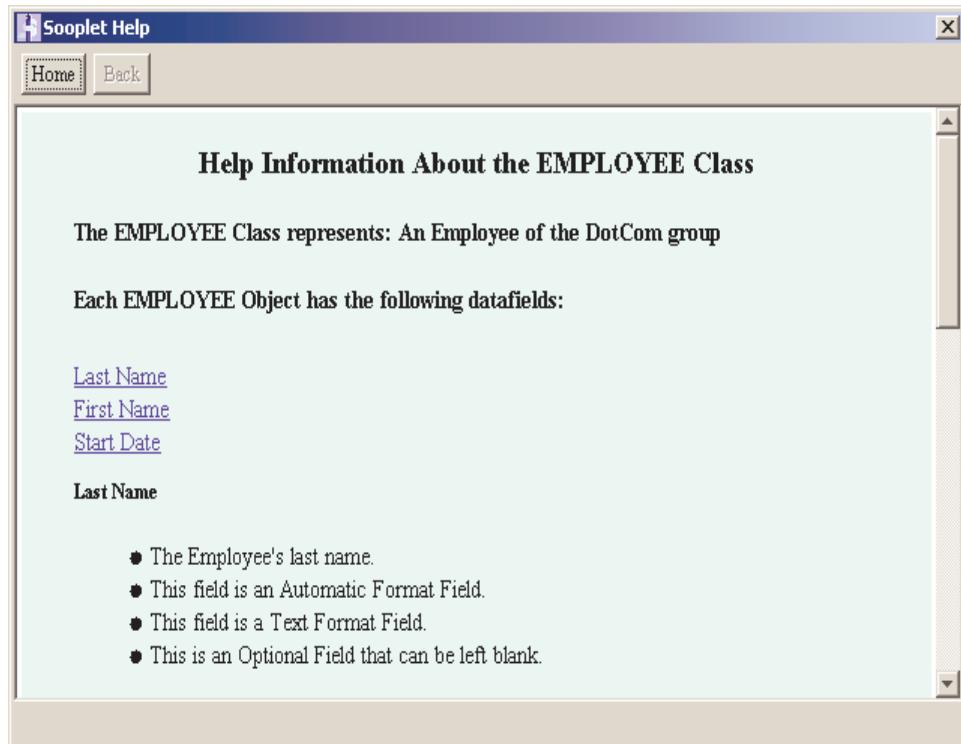
---

You will have noticed how the *Help* button appears in the Sooplet’s Table and Data Entry Form, and that it displays help about the particular Class when clicked. In this lesson we are going to enhance the help information, to better assist users at DotCom Industries to use our Sooplet.

1. In the diagram, choose *Edit Properties* from the EMPLOYEE Class menu. We are interested in the Description property. A default is provided, which we need to change to ‘**An Employee of the DotCom group.**’. Click *O.K.*
2. In the same way, change the Description for the POST Class to: ‘**A Post in the DotCom Structure.**’
3. Fields and Relationships also have a Description property. Right-click the Employee’s lastName Field and choose *Edit Properties* from the pop-up menu. Change the Description to: ‘**The Employee’s last name.**’. Click *O.K.*
4. Repeat for all Fields and Relationships in the Sooplet. The Relationship Properties are accessible from the Relationship menu, displayed when you right-click the name of a Relationship in the diagram, as illustrated.



5. Test your changes in the Sooplet's Data Entry View, by clicking the *Help* button on the lower right side, and also by pausing your mouse over column headings.



Lesson 7 is complete. The next lesson will cover self-relationships.

## Lesson 8 – Creating a Self-Relationship

---

Our Human Resources System Sooplet can now support Employees and their Posts. The next task is to establish ‘who works for whom’, i.e. model the organisational structure of DotCom. We do this by creating a Relationship from the POST Class to itself, representing a Post’s Manager Post.

1. In the Instant Soop diagram, open the POST Class pop-up menu and choose *Add Relationship*. The ‘Create Relationship’ dialog is displayed.
2. Name the relationship **reportsTo**
3. The Source Class is already selected; so choose POST as the target Class from the drop-down list.



Note how the ‘contains MANY’ option is disabled, as this is an illegal self-relationship.

4. Each Post has only one manager, so the default One-to-One Relationship is again appropriate.



The choice of Optional or Mandatory Constraint for this Relationship causes a minor dilemma. Consider the CEO, who has no manager. If we choose a Mandatory constraint, the CEO will have to manage herself/himself. If we choose an Optional constraint, we can leave the CEO’s manager blank, but this may lead to other posts missing managers and our structure will be inconsistent.

5. For now, we will use the Mandatory constraint so select this option.

In lesson 13 we will look at how inheritance can provide a better solution.

6. Click *O.K.* to create the Relationship. At this point, dismiss the dialog warning about Reverse Relationships, as we will add one later.

7. **Edit one of the Posts, assigning an appropriate Manager e.g. [DC1 CEO].**



Note how the reportsTo Post is represented by the Post Object Badge in the Data Entry View [DC1 CEO], and the Manager’s details can be displayed in a form by clicking the Badge.

In addition to ‘navigating’ from Employee to Post, we may also want to navigate from Post to Employee.



The Sooplet automatically maintains reverse relationships as we make connections between Objects in ordinary relationships. Once a connection has been made, it is not possible to add a reverse relationship, because the links would be rendered inconsistent. This is why we are warned about missing reverse relationships when we create a normal relationship. We are going to replace our ‘reportsTo’ Relationship with one supporting a Reverse relationship.

8. In the diagram, right-click the reportsTo Relationship to display the Relationship pop-up menu. Choose *Delete Relationship* and confirm the action.
9. Repeat steps 1 to 5 but before clicking *O.K.* tick the *Reverse Relationship* check box and rename the Reverse Relationship **manages**.

10. Edit each Post in turn, assigning an appropriate manager to *report to*.



Note how the selection of a reportsTo post, results in the subordinate post being added to the manages list of posts, for the manager concerned.

11. Change the reportsTo Post for an existing Post.

Confirm that the Post is removed from the former manager’s manages list of posts, and appears in the new manager’s manages list of posts.



This completes Lesson 8. In Lesson 9 we will see how to add fields that can perform calculations.

## Lesson 9 – Creating a Calculated Field

---

Fields can be configured to calculate their values based on the values of other fields. These are called ‘Calculated Fields’. In this lesson we will create a field for Annual Salary, and calculate the monthly salary from this.

1. Add a Field to the EMPLOYEE Class called **annualSalary**
2. From the Field pop-up menu for the annualSalary Field, choose *Set Data Type*. In the ‘Set Data Type’ dialog, clear the *Auto* check box and set the Data Type as a Number.
3. Add a second Field called **monthlySalary**
4. Set Prefixes for both fields to your currency symbol e.g. £ by editing the Field Properties.
5. From the Field menu for the monthlySalary Field choose *Set Data Type*.
6. Clear the **Auto** check box, set the type to **Calculated** and enter the formula **annualSalary/12** in the calculation box, and click *O.K.*.

Note that all Calculated Fields are set to Read-Only, to prevent users from editing the result.

7. Click *O.K.* to save the changes.
8. In your Sooplet, edit some Employee objects, adding values for Annual Salary before saving by clicking *O.K.*



Note the Calculator Widget that is provided to assist data entry of numbers, and how the Monthly Salary is calculated and displayed in the Employee Data Entry View.



This completes Lesson 9 on Calculated Fields. In Lesson 10 we will see how to add some custom behaviour to one of our Classes for users to execute.

## Lesson 10 – Creating a Method

---

A Method is a function within a Class, which operates on an object's data. You might need a Method for a sequence of operations in a business process, or a complicated calculation. Methods are written in SoopScript, which is a simple script language designed for the manipulation of Sooplet Classes, Fields and Objects.

In this lesson we are going to write a Method that assigns ownership of an Employee Object to the Employee herself/himself. We need to do this in the *next* lesson on Secure Sooplets, however we will build the method now to show how easy it is to use Methods to add functionality to a Class. Our first version will just display a message.

1. Right-click the **EMPLOYEE** Class and choose *Add Method* from the pop-up menu. The 'Add New Method' dialog will be displayed.

Our Method will be a 'Form Button' Method, which can be executed by a user clicking a button in the **Data Entry Form** for an Object. Ensure this is the type selected in the *Method Type* dropdown list.

2. Enter **Authorise** for the *Method Name*, and hit the Return key.



Note how a comment is automatically added as the first line of your method in the script pane:

```
//The authorise Form Button Method
```

The first line comment of the Method is important as it forms the description in Help pages.

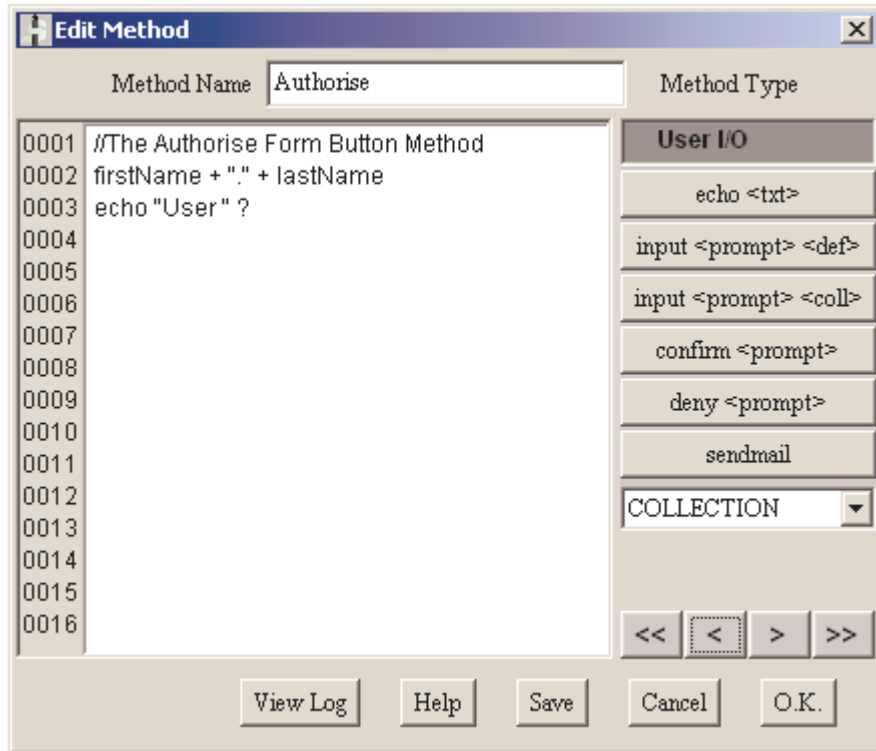
3. Add the following lines to your method



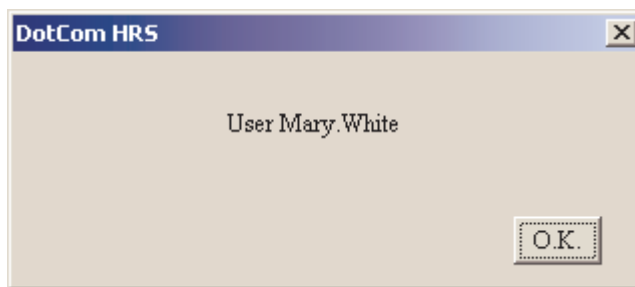
```
firstName + "." + lastName  
echo "User " ?
```

The *echo* command displays a message in a pop-up dialog, and the ? symbol represents the result from the previous command line, which in this case is the concatenation of the employee's names.

Your 'Method Editor' dialog should look like this:



4. Click the *O.K.* button to save your Method.
5. Edit an Employee in *Data Entry View* and you will see an *Authorise* button has been added, beneath the *Cancel* button.
6. Click this button and you will execute the method to display a message box with the message 'User Mary.White' or similar.



8. Click *O.K.* to dismiss the message, and *Cancel* to exit the form.

Having created our 'Form Button' Method, we now need to modify it to perform the task outlined above.

9. In the diagram, double-click the Authorise method in the EMPLOYEE Class. This will re-open the Method for editing.

10. Delete the last line and add the following two lines:



```
setowner #OBID# ?  
echo "The new owner is " #OWNER#
```

Your method should now look like the following:

```
//The Authorise Form Button Method  
firstName + "." + lastName  
setowner #OBID# ?  
echo "The new owner is " #OWNER#
```

The *setowner* command sets ownership of a named object to a specified user. The #OBID# property represents the internal id of an object, and the #OWNER# property represents an object's owner.

11. Click *O.K.* to save the Method.

We cannot test the enhanced Authorise Method at this stage, as Object Owners can only be changed in a Secure Sooplet.



The next lesson looks at securing a Sooplet, so by the end we will be able to authorise employees with our Method. Consult the **Soop Script Command Library** help for further information on scripting commands.

## Lesson 11 – Securing the Sooplet

---

Up until now, we have not concerned ourselves with the security of our Sooplet. No Logon was required, and all objects created were owned by 'Everyone'. This is not an ideal arrangement for DotCom Industries, who would like to impose some control over who does what in their HR System. In particular, the ability to create new employees should be restricted, but once created, an employee should be able to maintain her/his own object detail.



Roles and Permits can be established by the designer at design time, whilst Users, Roles and Permits can be configured by the Sooplet Administrator after deployment. We will look at design-time configuration first.

1. In Instant Soop, choose *Security* from the *Tools* menu. The 'Security Options' dialog will be displayed.
2. Check the *Enable Security* box.
3. Accept the default Administrator Username (theAdministrator), Administrator Password (blank) and Designer Password (blank), and click *O.K.* These values are fine for testing in this tutorial however you should consult the Help on Secure Sooplets for further advice, before deployment.



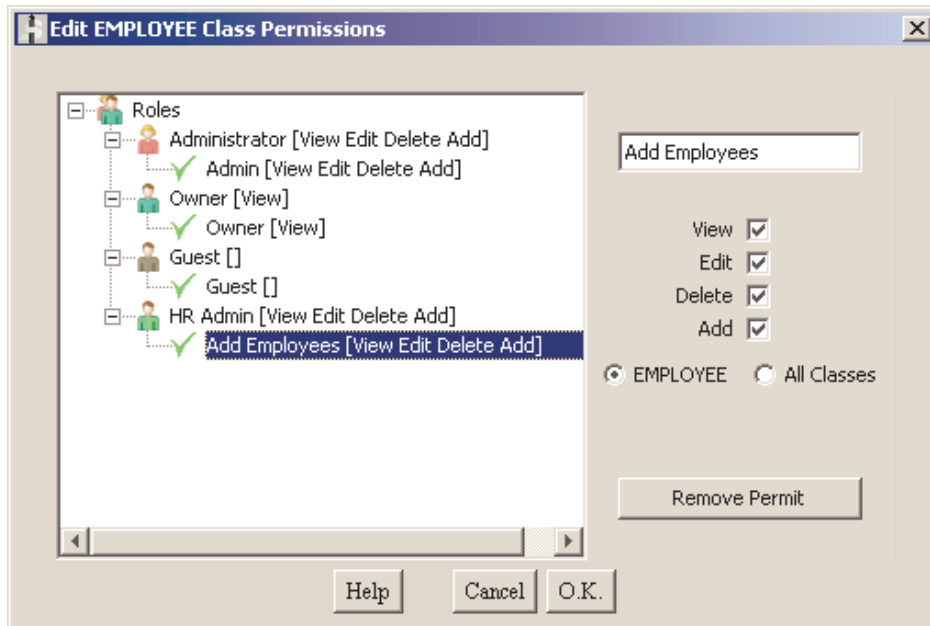
Head over to your secure Sooplet and note some changes. You should no longer be able to create new employees, as the permission to do this is denied. You can still edit and delete existing employees, as they were created in an unsecured Sooplet and are owned by 'Everyone'.

4. In the Instant Soop diagram, right click the EMPLOYEE class and choose *Edit Permissions*.

There should be two default roles displayed: Administrator and Owner.

5. Enter a new Role name **HR Admin** in the Role text box and click *Add Role*.
6. Enter a new Permit name **Add Employees** in the Permit text box and click *Add Permit*.
7. Check the *Edit*, *Delete*, *Add* and *View* boxes and choose the EMPLOYEE class.

Your permissions dialog should look similar to this:



8. Click O.K. to save your changes.

We now need to add a user to the Sooplet who will assume the role of HR Admin.

9. From the Sooplet *File* menu, choose *Logon*, select 'theAdministrator' from the drop-down list and click *O.K.*



Note that: the current user's username and role appears in the title bar of the Sooplet Window and a *Security* menu is available.

10. Choose *User* from the *Security* menu. Click *New User*, add a user of your choosing e.g. **HRManager** and click *O.K.*

11. Click the 'In Role' link for your new user, labelled '0 Userrole(s)', and assign them to the role of HR Admin by highlighting that row in the list. Click *O.K.*

12. From the Sooplet *File* menu, choose *Logon*, select your new user from the drop-down list, enter the password and click *O.K.*

13. Confirm that you can once again add new Employees, in your role as 'HR Admin'.

Following deployment, the Administrator role is the role that has the authority to allocate permission for different users to do different things in the Sooplet. We need to Logon as Administrator to set up these permissions.

14. From the Sooplet *File* menu, choose *Logon*, select theAdministrator from the drop-down list and click *O.K.*

15. Jot down the name of an employee, and from the Sooplet *Security* menu choose *User*, click *New User* and add the employee as a User. The Username should be constructed accurately from the user's first name and last name, separated by a dot, e.g. **Fred.Smith** and the password can be any memorable phrase – **fred** will suffice for testing. Repeat this for each remaining employee.



Note that we don't need to specify a role for each employee, as we are making use of the 'Owner' role, which employees automatically assume for objects they own.

16. From the *Security* menu choose *Role* and in the Owner Role Object click over '1 Ownerpermit [...]' to view the Ownerpermits. Add a new Ownerpermit and set it up as follows, named **EmployeeAutonomy**, Can View = Yes, Can Edit = Yes, Can Delete = No, Class = *EMPLOYEE*.

Your form should look like the one below:

The screenshot shows a web browser window titled "DotCom HRS Data Edit [theAdministrator as Administrator]". The menu bar includes "File", "Data Entry", "Data Query", "Reports", "Documents", "Security", and "Help". The main content area is titled "Edit Details for new Ownerpermit" and contains the following form fields:

- Name:** A text input field containing "EmployeeAutonomy".
- Can View:** Radio buttons for "Yes" (selected) and "No".
- Can Edit:** Radio buttons for "Yes" (selected) and "No".
- Can Delete:** Radio buttons for "Yes" and "No" (selected).
- Class:** Radio buttons for "[ALL]", "[EMPLOYEE]" (selected), and "[POST]".

On the right side of the form, there are three buttons: "O.K.", "Cancel", and "Help". At the bottom right of the window, the URL "www.Sooplet.com" is displayed.

17. Click *O.K.* to add the permit.
18. *Edit* the Employee Object for your first employee, and use the *Authorise* button to delegate ownership of the Object. Repeat this step to authorise each Employee.



You should now be able to select each Employee Object, and observe that its owner is the user associated with that Employee. An object's owner is

displayed in the lower left-hand status panel in Data Entry View.

19. Logon as your first employee and check that they can View and Edit their own details but not those of other Employees. Also check they can't create employees or posts and can't delete any employees. Check that this is true for all employees by logging on as each one in turn.



This completes Lesson 11, where we have explored the facilities afforded by a Secure Sooplet. In Lesson 12 we will place shortcuts for users to work with our Sooplet.

## Lesson 12 – Placing Shortcuts

---

We have developed sufficient functionality and security within our Sooplet to deploy it to the HR Department. Sooplet users will not be interested in the design diagram, or any features of Instant Soop, indeed we would wish to hide such detail.



To achieve this we can publish shortcuts on the Desktop and *Start* menu, which will open the DotCom HRS Sooplet as a standalone application. Please note this feature relies on functionality found in Windows platforms. Users on other platforms should consult the Help on shortcuts for alternative solutions.

1. From the Instant Soop *Sooplet* menu choose *Create Shortcuts*
2. Answer *Yes* when asked to confirm your choice.

View the PC Desktop and confirm that a shortcut has been placed



Navigate the *Start* menu and confirm an entry for DotCom HRS has been placed in the Sooplets Program Group.

3. Close your test Sooplet and Exit from Instant Soop using the *Exit* option of the Instant Soop *File* menu.
4. Click on the Desktop Shortcut and confirm the DotCom HRS Sooplet is opened.



Note that a splash screen is displayed with the title of your Sooplet, and within the logon dialog the username is no longer provided in a drop-down list as it was in the test environment.

5. Close the Sooplet and try to re-open from the *Start* menu shortcut. Close the Sooplet.



That completes Lesson 12. If you have made it this far in one sitting, you deserve a break before tackling the next lesson on inheritance!

## Lesson 13 – Inheritance

---



Inheritance is a powerful technique for enabling one class, the *child* class, to inherit functionality from another class, the *parent* class. This reduces duplication in the design and makes it easier to update. We can employ inheritance to resolve our dilemma from lesson 8. You may recall that the CEO has no manager, but we were unable to represent this fact. What we will do now is create a new parent class for the CEO, and make the POST class inherit from it.

Re-open the Instant Soop designer and the DotCom HRS Sooplet.

1. Delete the *CEO POST* Object in your Sooplet by selecting it and clicking the *Delete Post* button.
2. Delete the reportsTo relationship by right-clicking it in the diagram and choosing *Delete Relationship*.
3. From the Instant Soop *View* menu choose *Inheritance*.

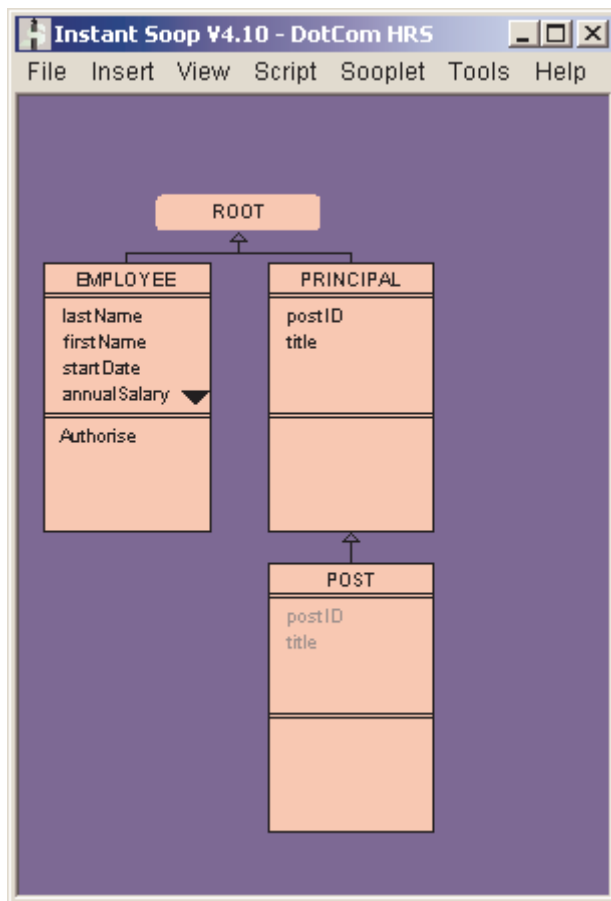
This view displays the class hierarchy i.e. how each class is linked via inheritance. All the menu options available in *Association* view are also available in *Inheritance* view.

4. Right-click the diagram and choose *Add Class*.
5. Enter **PRINCIPAL** as the *Class Name* and then click *O.K.*
6. Drag the POST Class over the PRINCIPAL Class to make it a child of PRINCIPAL.

The POST class will move under the PRINCIPAL class with an up-arrow joining the two.

7. Drag the postID field from the POST Class to the fields section of the PRINCIPAL class. Repeat with the title field.
8. Set the badge for the PRINCIPAL class to [postID title] as you did in lesson 5.

Your diagram should now look like the following:



Note that inherited fields are shaded grey, and in your Sooplet the original POST data has been retained, even though POST fields are now inherited from PRINCIPAL.

9. Switch to the *Association View*, and re-create the One-to-One reportsTo Relationship, from POST to PRINCIPAL.
10. Logon as administrator and create a new CEO object by choosing *PRINCIPAL* from the Sooplet *Data Entry* menu, and clicking *New Principal*. Choose CEO as the Title from the drop-down list of previous entries and click O.K.



Note that you are not required to enter a manager for the CEO.

11. Replace the Employee's thePost relationship with an identical relationship, but this time make the target the PRINCIPAL Class, instead of the POST Class.



Note that as POST inherits from PRINCIPAL, Posts are effectively treated as Principals. This means that when we associate EMPLOYEE with PRINCIPAL, we can choose Posts or Principals for the 'thePost' relationship.

12. Edit your previous POST objects, adding appropriate entries for their managers.
13. Edit your EMPLOYEE objects, choosing appropriate entries for their jobs.



Lesson 13 is complete. See the Help for further details on Inheritance. Querying data is the next topic.

## Lesson 14 – Querying Data

---

DotCom HR staff may be called upon to answer questions about DotCom Industries, such as ‘How Many Employees’, etc. To do this we will need to Query Data held in the Sooplet. Sooplets provide a simple method for querying data.

Queries are based on a single Class, or a ‘route’ through the Classes via chosen relationships. Fields that we do not need to show can be hidden. Objects that we do not require can be filtered out.

We are going to build a query that lists Employee details for new starters, based on a route from EMPLOYEE to POST via thePost relationship.

Logon as theAdministrator to gain full access to the Sooplet.

1. From the *Data Query* menu choose the **Employee** ▶ submenu
2. From *Employee* submenu choose **thePost PRINCIPAL** option

This will provide a results table based on Employees and their jobs.

3. Hide the monthly salary and CLASS columns by right clicking the columns and choosing Hide Column from the Column menu.
4. Filter the Employees for those with Start Dates after today by entering **>t** in the criteria box above the EMPLOYEE\_startDate Column. Other filter options are detailed in the Help files.



Note how this reduces the number of employees displayed in the query.

5. From the *File* menu choose *Save Query*. Name the query ‘**New Starters**’ and select ‘Make Available to Everyone’. Click *O.K.* to save the query.
6. Experiment with the Data Query facilities to ensure the query view has changed.
7. From the *Data Query* menu’s Public Queries submenu, choose ‘*New Starters*’ and confirm that your saved query is recalled.



That completes Lesson 14. We will use our query again in lesson 16 as the basis for a welcome letter.

## Lesson 15 – Generating Reports

---

Reports are snapshots of Sooplet data, available in a variety of formats. Reports are based either on Routes, as Queries (see previous lesson), or Relations of a particular Class e.g. Employee.\*

1. Logon to the Sooplet as the Administrator, if you have not already done so.
2. From *Reports* menu choose **Routes** ▶.
3. From *Routes* submenu choose **EMPLOYEE** ▶
4. From *Employees* submenu choose the **thePost PRINCIPAL** option.
5. Accept the default *csv* format option, and click *O.K.* when the ‘Run Report’ dialog is displayed. The report will appear in your default Browser.
6. Repeat the above with various options from the *Route* and *Relation* menus, in various formats.



Reports are created in the DotCom HRS subfolder of the reports folder, under the default Instant Soop data folder.



This completes Lesson 15. We will now look at creating a document based on data from our Sooplet.

## Lesson 16 – Creating a Document

---

Sooplet Documents can be reports, letters, or any page-based publication based on a template merged with Sooplet Data. The document format is Rich Text Format (RTF), so Documents can be created and viewed by any RTF compatible Word Processing tool that supports merge fields e.g. Microsoft Word.

In this lesson we will create a welcome letter that is merged with data from our New Starters query that was created in Lesson 14.

1. From the Sooplet's *Data Query* menu, open the *New Starters* query that will be found in the *Public Queries* submenu.
2. From the *File* menu choose *Merge Query with Document*
3. Enter the document name **Welcome Letter**, accept the default option to *Keep Private* and click *O.K.* to create the document.

At this stage the Word Processor will be launched with a new document containing the following instructional text and sample fields:

### Instructions for Document Creation:

1. Use your Word Processor's list of Merge Fields to insert a new merge field.

Some sample fields have been provided below.

Some fields may be child lists that should be horizontally aligned in tables or columns.

2. Delete these instructions and enter your template text to create the master document.

3. Save the completed document WITHOUT changing the filename or location.

4. Activate the merge to create your production output.

```
EMPLOYEE_lastName: <<EMPLOYEE_lastName>>
EMPLOYEE_firstName: <<EMPLOYEE_firstName>>
EMPLOYEE_startDate: <<EMPLOYEE_startDate>>
EMPLOYEE_annualSalary: <<EMPLOYEE_annualSalary>>
PRINCIPAL_postID: << PRINCIPAL _postID>>
PRINCIPAL _title: << PRINCIPAL _title>>
```



If the default Instant Soop configuration fails to locate your Word Processor, then the configuration file will need to be amended. See Instant Soop Help on Configuring Instant Soop Properties.

4. Edit the document into a suitable welcome letter incorporating the merge fields as required. An example is shown below, which you can cut, paste and add your merge fields to if required:

```
FAO «EMPLOYEE_firstName»          DOTCOM Industries
«EMPLOYEE_lastName»                Ivory Tower
                                      Baker Street
                                      Downtown
                                      HR IS1

Dear «EMPLOYEE_firstName»

Following your successful appointment to the post of
«PRINCIPAL_title» at DOTCOM Industries we would like to
welcome you to the organization, and confirm a few details
about your appointment.

Your annual salary will be «EMPLOYEE_annualSalary».

We look forward to seeing you at 0900 on
«EMPLOYEE_startDate».

Yours sincerely

A R Scott

HR Manager
```

5. Save the document without changing the filename or file location. Close the document in your RTF editor.
6. Edit some employee details of a new starter e.g. the annual salary.
7. To produce the document again, with fresh data, open the document from the Sooplet's *Documents* menu.



This completes Lesson 16. We have created a template Document that can be generated for any new starters that join DotCom Industries. In the next lesson we will explore the scripting facilities of Instant Soop.

## Lesson 17 – Saving Sooplet as Script

---

Instant Soop has a powerful scripting language to enable Sooplets, or individual Classes to be saved to a single file. This is useful for distributing designs, making back-ups and creating test applications.

1. From the Instant Soop *Script* menu choose *Save Sooplet*.
2. Accept the default name of *DotCom HRS*.
3. Click *O.K.* to save script.

DotCom has been acquired by Amaz Books.

4. Create a new Sooplet called Amaz Books (refer to Lesson 1 if necessary).
5. Run the *DotCom HRS* Script, by choosing it from the list of recent scripts at the bottom of the *Script* menu.



Note the DotCom HRS Sooplet has been duplicated complete with its data and queries, but no documents or security configuration.



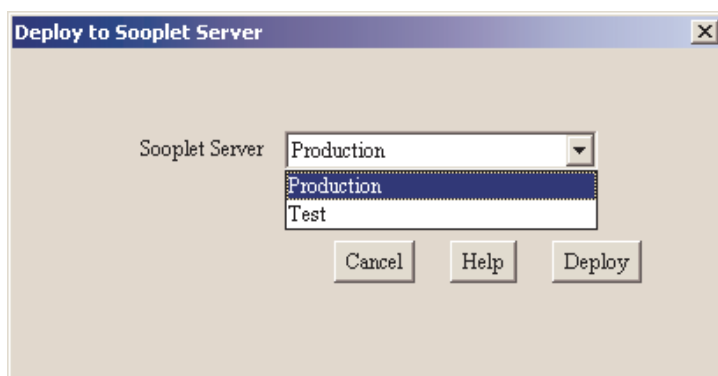
This completes Lesson 17. In the final lessons we will look at 2 ways to deploy our Sooplet.

## Lesson 18 –Deploying a Sooplet to SoopletServer

---

Deploying Sooplets for others to use is extremely easy. This lesson looks at deploying to the Sooplet Server, a separate application available from the Sooplet.com website, which makes your Sooplet available as a fully interactive web application and a Client/Server application via the dedicated client. The final lesson will cover deploying to webservers for wider distribution.

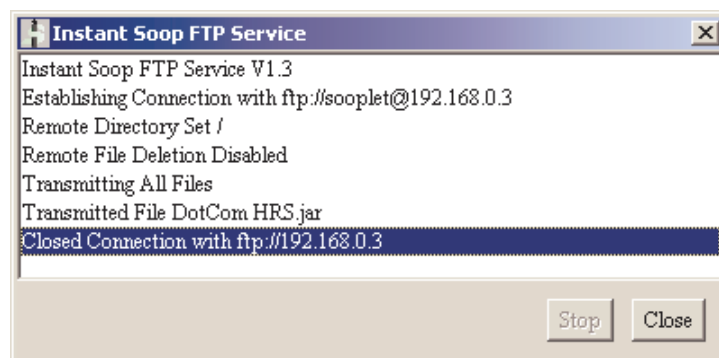
1. Download and install Sooplet Server from <http://www.sooplet.com/downloads>
2. Re-open the DotCom HRS Sooplet by choosing from the recently-used list on the Instant Soop *File* menu. From the *Sooplet* menu choose *Deploy to Sooplet Server*. The following dialog will be displayed:



Note that the Instant Soop Designer can automatically discover Sooplet Server installations from your network.

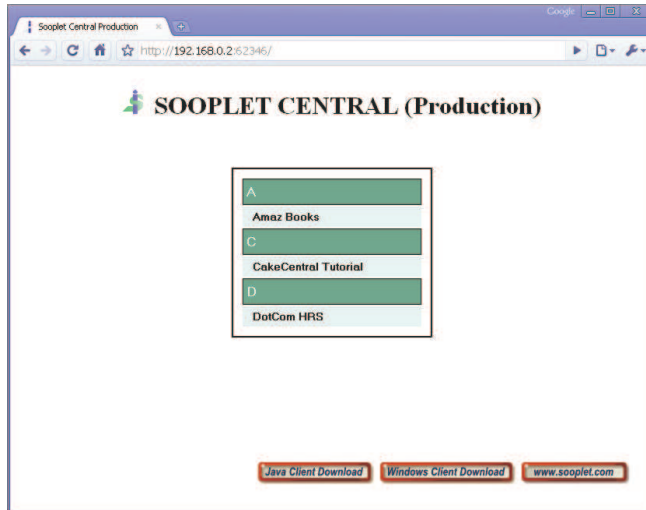
3. Choose which Sooplet Server you wish to use from the drop-down list.
4. Click the *Deploy* button.

The progress of file transmission will be displayed in the 'FTP Service' dialog, as shown.



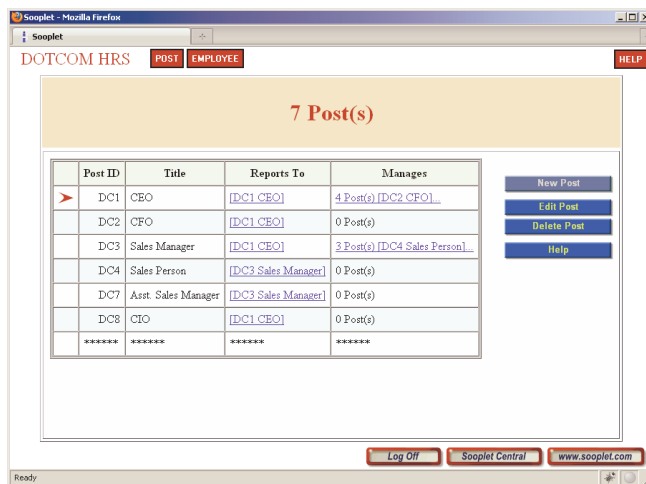
- When deployment is complete close the 'FTP Service' dialog, and accept the success message.
- In your favourite Web Browser navigate to the Sooplet Central Home Page. This will be found at the following URL for a default Sooplet Server install on the same machine: <http://localhost:62346>

The Sooplet Central home page will be displayed as shown:



- Choose *DotCom HRS* from the A-Z listing of available Sooplets.

The DotCom HRS Sooplet will appear in your browser, and you can administer employees and posts in the familiar way.



- Log Off.



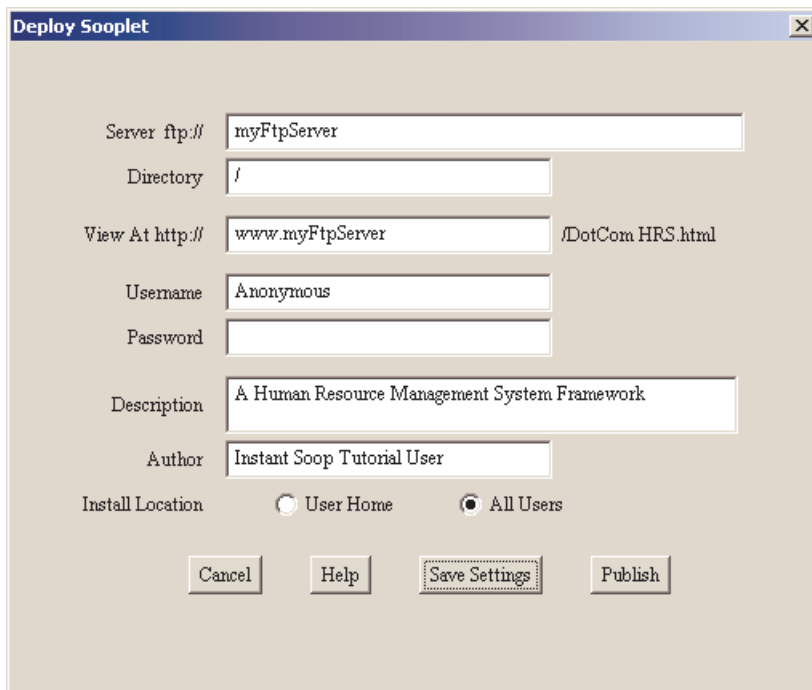
This completes lesson 18.

## Lesson 19 –Distribute a Sooplet via the Web

---

Deploying your Sooplet for users beyond your local network is also straightforward. All the files you require are assembled and transmitted in a single operation, including the generation of a 'Download' page introducing your application. You will need access to a Webserver which must be capable of receiving files using the File Transfer Protocol (FTP), or you can make use of the 'Gallery' free hosting facility at Sooplet.com [www.sooplet.com/gallery](http://www.sooplet.com/gallery)

1. From the Instant Soop *Sooplet* menu choose *Deploy Sooplet*. The following dialog (shown with sample data) will be displayed:



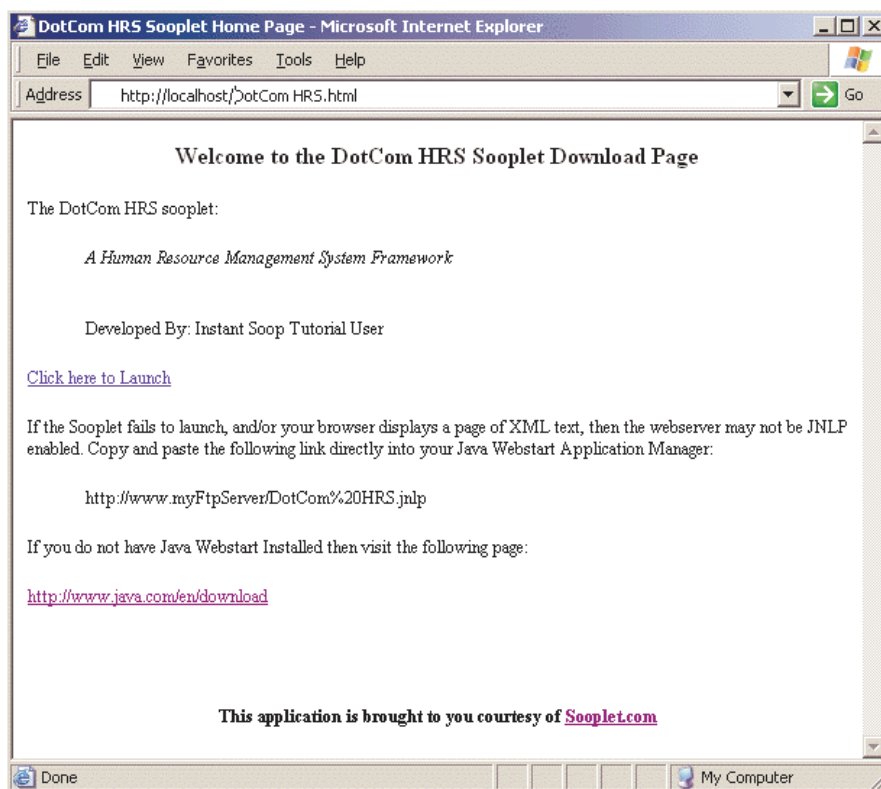
The screenshot shows a dialog box titled "Deploy Sooplet" with the following fields and options:

- Server ftp://: myFtpServer
- Directory: /
- View At http://: www.myFtpServer /DotCom HRS.html
- Username: Anonymous
- Password: (empty)
- Description: A Human Resource Management System Framework
- Author: Instant Soop Tutorial User
- Install Location:  User Home  All Users
- Buttons: Cancel, Help, Save Settings, Publish

Note the default connection details when deploying a new Sooplet will be for the Sooplet.com Gallery, not the example ones shown above.

2. Enter the URL of the FTP Server as specified by your Internet Service Provider (ISP) or Systems Administrator, and any Username and Password required for access.
3. Specify the Directory on the server where you want to store your application. This directory will be created if it doesn't exist.
4. By entering the server address in the *View At URL*, you will be able to view the deployment site's download page following file transmission.

5. Enter a useful description for your Sooplet, and your name/company as *Author*.
6. The install location dictates whether the deployed application, when installed, is installed in the users home folder or made available to all users.
7. The next step is to ensure connectivity to the target webserver (by connecting to the internet if necessary) before clicking the *Publish* button.
8. The progress of file transmission can be monitored in the 'FTP Service' dialog, as shown in the previous lesson.
9. When you have successfully deployed your Sooplet, dismiss the 'FTP Service' dialog by clicking the *Close* button. You will then be offered an opportunity to view the Download Page for your Sooplet application – which should resemble the following:



This completes the final lesson...



You have completed the DotCom HRS Sooplet Tutorial.

