

INSTANT SOOP
DESIGNER'S GUIDE

Version 4.1

Table Of Contents

1	The Instant Soop Designer	1
1.1	Installing the Instant Soop Designer	1
2	Creating a New Sooplet.....	1
3	Opening a Sooplet.....	2
4	Sooplet Classes	2
4.1	Creating a New Class	2
4.2	Identifying Sooplet Classes	3
4.3	Deleting a Class.....	3
4.4	Adding a Child Class.....	3
4.5	Inserting a New Class From a Database	3
4.6	Class Properties.....	4
4.7	Delete All Classes	5
5	Sooplet Fields	5
5.1	Creating a New Field.....	5
5.2	Identifying Sooplet Fields.....	6
5.3	Deleting a Field	6
5.4	Renaming a Field	6
5.5	Field Properties	6
5.5.1	Prefix.....	7
5.5.2	Suffix	7
5.5.3	Field Description	7
5.5.4	Header	7
5.5.5	Footer	7
5.5.6	Options	8
5.6	Moving and Copying Fields.....	8
6	Sooplet Relationships.....	9
6.1	Creating a New Relationship	9
6.1.1	Naming Relationships.....	10
6.1.2	Choosing the Source and Target Class	10
6.1.3	Selecting the Mode	10
6.1.4	Constraints Settings	12
6.1.5	Advanced Relationship Options: Reverse Relationships.....	12
6.1.6	Advanced Relationship Options: Sub-Targets	13
6.1.7	Advanced Relationship Options: Filters	13
6.1.8	Editing Relationships.....	13
6.1.9	Renaming Relationships	13
6.1.10	Deleting Relationships.....	14
6.2	Relationship Properties.....	14
6.2.1	Description	14
6.2.2	Header	14
6.2.3	Footer	14
7	Sooplet Methods	15
7.1	Creating a New Method.....	15
7.1.1	Table and Form Button Methods.....	16
7.1.2	Object Creation Method.....	16
7.1.3	Data Edit Method.....	16
7.1.4	Data OK Method	16
7.1.5	Private Method	16

7.1.6	Hidden Method.....	16
7.1.7	Pulse Method.....	17
7.2	Deleting a Method.....	17
7.3	Renaming a Method.....	17
7.4	Editing a Method.....	17
7.5	Splitting a Command over multiple Lines.....	17
7.6	Moving and Copying Methods.....	17
8	Sooplet Data.....	18
8.1	Delete All Data.....	18
9	Context Menus.....	18
9.1	The Sooplet Design Menu.....	18
9.2	The Class Menu.....	19
9.3	The Field Menu.....	19
9.4	The Method Menu.....	19
9.5	The Relationship Menu.....	19
9.6	The Reverse Relationship Menu.....	19
10	Preview Sooplet.....	20
11	Creating a Desktop Shortcut.....	20
11.1	Setting a Shortcut Manually (Windows).....	21
11.2	Setting a Shortcut Manually (Java Webstart for Windows).....	21
11.3	Setting a Shortcut Manually (Java Webstart for Linux).....	21
12	Deploying a Sooplet.....	21
12.1	Deploying Sooplet as a remote Java Webstart Application.....	22
12.2	Deploying Sooplet as a local Java Webstart Application.....	22
12.3	Deploying Sooplet to a Sooplet Server.....	22
12.3.1	Auto-Deploy.....	23
12.3.2	FTP Publishing.....	23
12.3.3	Manual Copy.....	23
13	Sooplet Security Options.....	24
13.1	Enabling Security.....	24
13.2	Class Permissions.....	25
14	Soop Scripts.....	27
14.1	Running a Script.....	27
14.2	Save Class As Script.....	27
14.3	Save Sooplet As Script.....	28
15	Printing a Sooplet.....	28
15.1	Print Preview.....	28
16	Configuring Instant Soop Properties.....	29
16.1	RTF Document Settings.....	29
16.2	Most Recently Used File Lists.....	30
16.3	Scripts.....	30
16.4	Sooplet Help.....	30
16.5	Regional Settings.....	30
16.6	Deployment Settings.....	31
16.7	EMail Settings.....	31
16.8	Pulse Period.....	31
17	Configuring Sooplet Properties.....	31
17.1	Form Settings.....	31
17.2	Reports.....	32
17.2.1	Prefixes and Suffixes.....	32

17.2.2	Relationship Names	32
17.3	Deployment.....	32
17.4	Object Locking.....	32
17.5	Pulse Period	33
18	Closing a Sooplet	33
19	Exiting Instant Soop.....	34
20	Help	35
20.1	Instant Soop Help Index	35
20.1.1	Menubar Menu Options	35
20.1.2	PopUp Menu Options	35
20.2	GLOSSARY	36

1 The Instant Soop Designer

The Instant Soop Designer is the tool used to create Sooplets. Sooplets are complete self-contained, data-based applications, designed and deployed to a web or client-server environment in ultra-rapid development cycles.

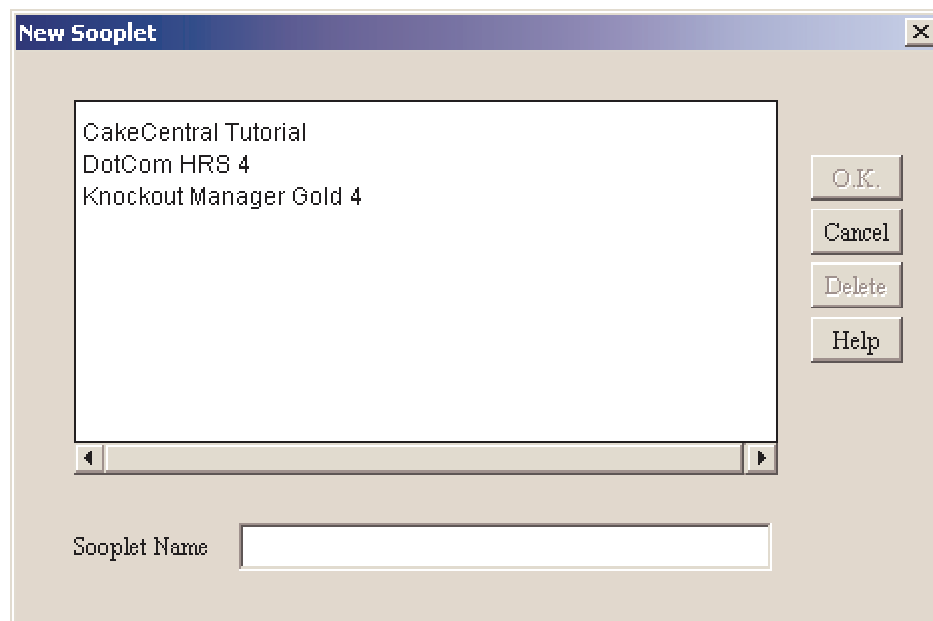
This manual documents the features of Instant Soop Version 4.1. There are other documents which you may find useful: 'The Sooplet User Guide', which covers using a Sooplet, the 'DotCom HRS Tutorial' which provides a step-by-step guide to creating your first Sooplet, and the 'SoopScript Command Language Guide' for a detailed description of each script command.

1.1 Installing the Instant Soop Designer

The Instant Soop Designer can be downloaded from <http://www.sooplet.com/downloads>

2 Creating a New Sooplet

To create a new Sooplet choose New from the File Menu. The following Dialog will be displayed:



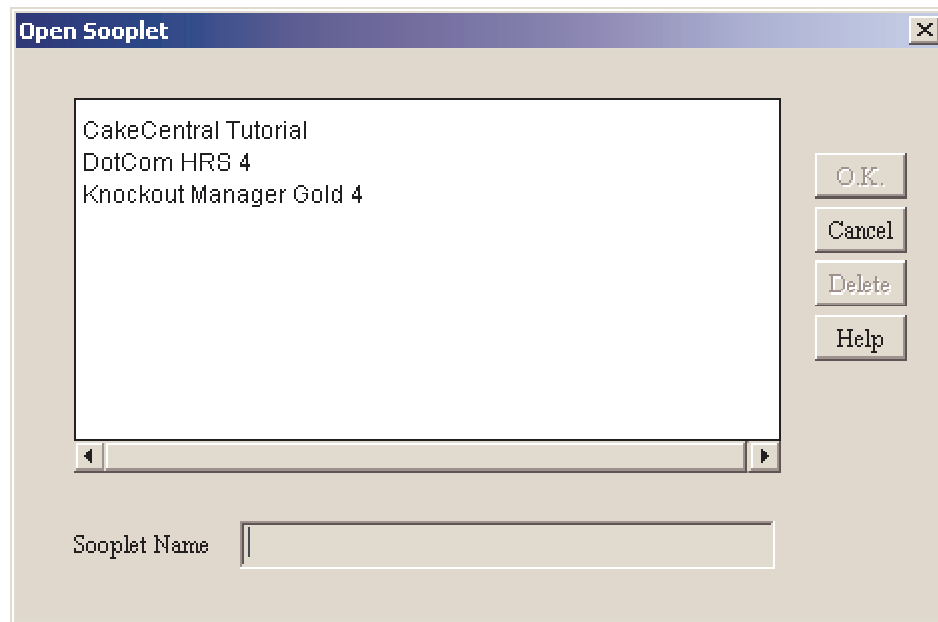
Enter a name in the 'Sooplet Name' box and click OK. Sooplet names are limited to 24 characters and some characters are restricted. To adapt an existing name, first choose from the list and then edit the name in the name box before clicking O.K. Sooplets can also be deleted by selecting from the list and clicking the Delete button.

See Also

[Opening a Sooplet](#)

3 Opening a Sooplet

To open an existing Sooplet choose Open from the File Menu. The following Dialog will be displayed:



Choose a Sooplet from the list and click OK, or double-click on a Sooplet in the list. Sooplets can also be deleted from this dialog by selecting from the list and clicking the Delete button.

See Also

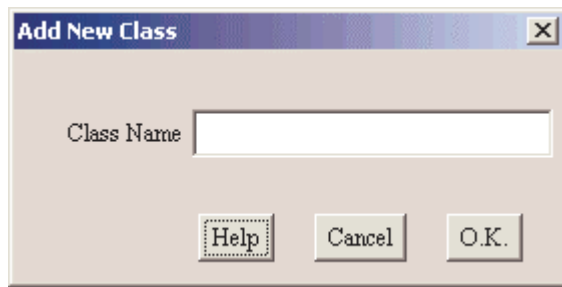
[Creating a Sooplet](#)

4 Sooplet Classes

4.1 Creating a New Class

Classes are the things that represent significant elements in your Sooplet design. Class names are restricted to 16 characters in length, must not start with a numeric character, and will be automatically converted to Upper Case text. Classes should be singular: for example CUSTOMER, ORDER, INVOICE, etc. Some class names may be rejected due to clashes with Soop reserved words, in which case an 'Error in Class Name' message will be displayed.

Choosing 'New Class' from the Insert menu, or 'Add Class' from the pop-up menu that appears when you right-click the Sooplet Diagram, will present the following dialog:



4.2 Identifying Sooplet Classes

There are various techniques for identifying the classes that might be needed in a particular application. One technique involves highlighting all the nouns in the 'Statement of Requirements' that describes the problem area. However, as Instant Soop builds the Sooplet Application immediately, a less formal approach to Class identification can be adopted. Changes to the design are instantly reflected in the built application, promoting a more iterative approach to development. In addition, data can be entered very early on in the development process, which is often the point at which the design reveals its weaknesses. The recommendation therefore, is to create skeleton classes, with minimal fields, to enable the design to be tested at an early stage.

4.3 Deleting a Class

Choosing 'Delete Class' from the Class Menu will delete the Class and all data associated with Objects of that Class. You will be prompted to confirm your action.

4.4 Adding a Child Class

Choosing 'Add Child Class' from the Class Menu will add a new Class that inherits from this Class. You can also drag one Class over another in the 'Inheritance View' to implement inheritance. To reset inheritance, drag your class over the ROOT class.

See Also

[Add Class From Database](#)

[Class Properties](#)

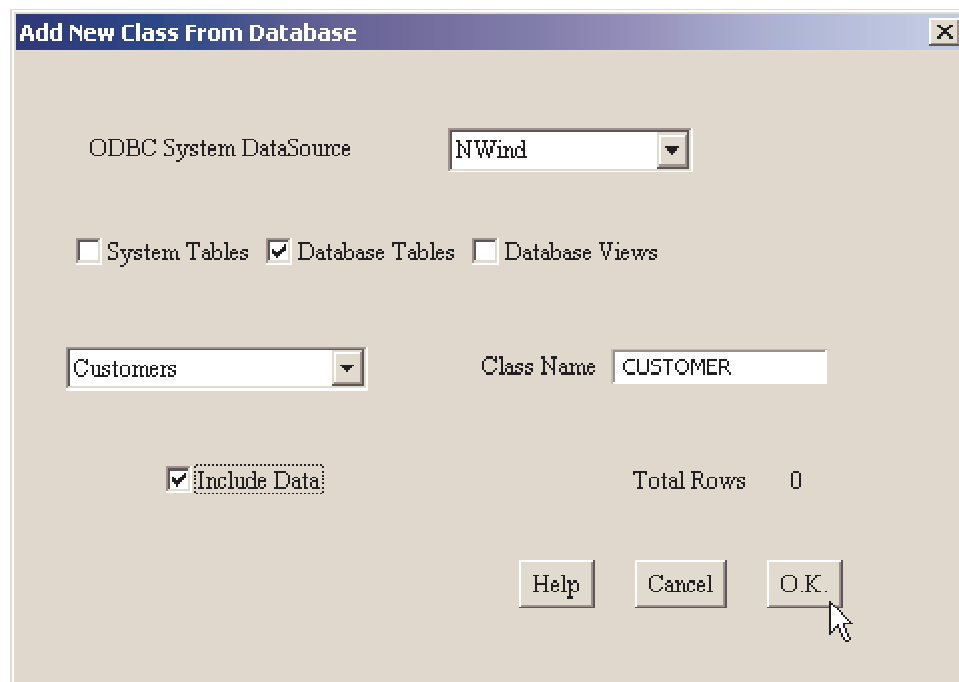
[Inheritance](#)

[Changing Views](#)

4.5 Inserting a New Class From a Database

Classes can be created from the structure of a relational database table or view. In addition data can be imported, with each row from the table becoming an object in the Sooplet. This facility requires an ODBC System DataSource to be set up pointing to

the target database. As a result this facility may not be available on all platforms. If you choose 'Class from Database' from the Insert Menu the following dialog will be displayed.



The source database is chosen from the drop-down list of ODBC System DataSources. Some platforms may not support Auto-discovery of ODBC Datasources, in which case a Text Box will be provided to enter the DataSource Name and a Fetch Button to action the request.

The available tables can be filtered by ticking the following options: System Tables, Database Tables and Database Views.

Having selected datasource and filter options, you can choose a table from the 'Choose Table' drop-down list. A suggested Class Name will be displayed which can be edited if required. If data is required, as well as the Class structure then tick the 'Include Data' checkbox. Click the OK Button to start the insertion.

The progress of the operation will be displayed as a count of rows.

See Also

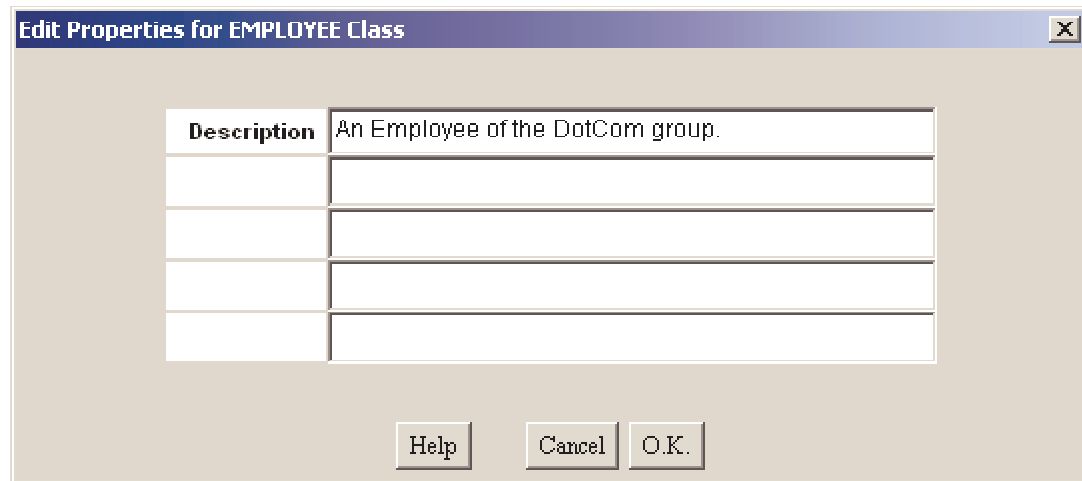
[Create New Class](#)

4.6 Class Properties

The Class Properties dialog provides the facility to edit the 'Description' of a Class. The Class Description is a segment of text that should briefly describe the purpose of the Class, and its role within the Sooplet. For example, a useful description of an EMPLOYEE Class might be:

The EMPLOYEE Class represents an Employee of the Company.

The Class Properties dialog can be displayed by choosing Edit Properties from the Class Menu.



The Class Description will appear in status text and help text throughout the Sooplet.

See Also

[Field Properties](#)

[Relationship Properties](#)

4.7 Delete All Classes

Choosing 'Delete All Classes' from the Sooplet Menu will delete all the classes in your Sooplet, including all objects, collections, data, queries and documents! Be sure this is what you want to do, before answering 'Yes' to the warning prompt.

See Also

[Delete All Data](#)

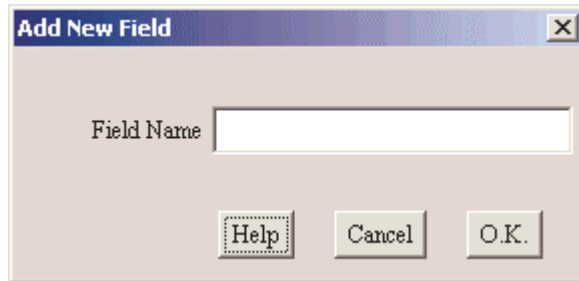
5 Sooplet Fields

5.1 Creating a New Field

Fields are the attributes of a Class that hold data about an object. Field names are restricted to 16 characters in length, and can comprise upper or lower case characters, digits and some punctuation. Spaces are not permitted in Field names however, composite names with the first character of the second word capitalised, such as

firstName and lastName, will be presented as 'First Name' and 'Last Name' in the Sooplet. Some Field names may be rejected due to clashes with Instant Soop reserved words, in which case an 'Error in Field Name' message will be displayed.

Choosing 'Add Field' from the Class Menu will display the following dialog:



5.2 Identifying Sooplet Fields

There are various techniques for identifying the Fields that might be relevant to a particular Class. There are basically two approaches. One involves creating the Classes and their Relationships first, before designing the Class's Fields in detail. The other approach uses an analysis of the problem area to determine the attributes required for each Class. Either approach is suited to Instant Soop, as unlike a traditional coded development, Fields can be moved and copied between Classes with the drag of a mouse. In addition, there is no User Interface to be modified following such changes, so the design can be constructed incrementally in a more intuitive fashion.

5.3 Deleting a Field

Choosing 'Delete Field' from the Field Menu will delete the Field and all data associated with that Field. You will be prompted to confirm your action.

5.4 Renaming a Field

Choosing 'Rename Field' from the Field Menu will display a dialog to enable the Field name to be changed. In this case no data will be deleted.

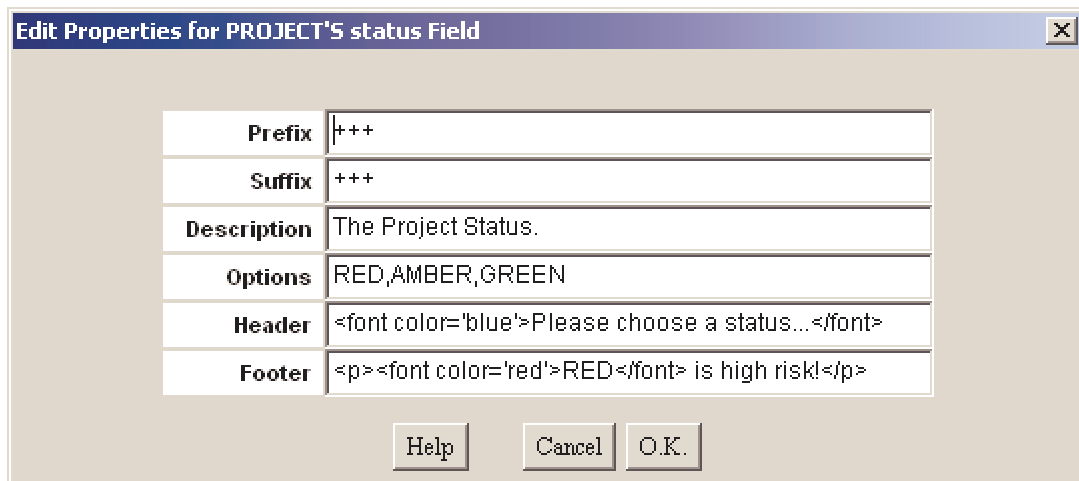
See Also

[Moving and Copying Fields](#)

[Creating a New Method](#)

5.5 Field Properties

The Field Properties dialog provides the facility to edit the following properties: the Prefix, Suffix, Description, Header, Footer and Options. The Field Properties dialog can be displayed by choosing Edit Properties from the Field Menu.



5.5.1 Prefix

The Prefix is a snippet of text that will precede all data values in this field. For example, a field holding currency values may be prefixed with \$ or £.

5.5.2 Suffix

The Suffix is a snippet of text that will be appended to all data values in this field. For example, a field holding percentages values may be suffixed with %.

Prefixes and Suffixes are employed for display purposes only, so will appear in Badges, but are ignored by the search and filter facilities provided in Data Entry, Data Query and Data Selection views.

5.5.3 Field Description

The Field Description is a segment of text that briefly describes the purpose of the Field.

The Field Description will appear in status text and help text throughout the Sooplet.

5.5.4 Header

The Header is a configurable block that will sit above this field in the Data Entry Form. For example, advice for entering values may be placed in the header.

5.5.5 Footer

The Footer is a configurable block that will sit below this field in the Data Entry Form. For example, dividing lines may be placed in the footer.

Headers and footers can contain simple html markup, e.g. Please choose... [Please choose...](#)

5.5.6 Options

The Options property enables a comma separated list of values to be defined, that forms the options from which a selection can be made. For example, to provide a traffic light status you might define the options RED,AMBER,GREEN. This would appear in the sooplet as follows:

The screenshot shows a dialog box titled "ProjectManagerAssistant Data Edit" with a menu bar containing "File", "Data Entry", "Data Query", "Reports", "Documents", "Security", and "Help". The main content area is titled "Edit Details for [Sooplet Marketing]". It features three input fields: "Name" (containing "Sooplet Marketing"), "Start Date" (containing "01/12/09"), and "End Date" (containing "31/12/15"). Below these is a "Status" field with "AMBER" selected, flanked by "+++" on both sides. A message "Please choose a status..." is positioned above the status field, and "RED is high risk!" is below it. On the right side, there are three buttons: "O.K.", "Cancel", and "Help". At the bottom, there is a footer with "The Project Status." and the website "www.sooplet.com".

In the above example, users can only select from the list. They cannot add new values. To make the list extensible, add a further comma to the end of the options e.g. RED,AMBER,GREEN,

See Also

[Class Properties](#)

[Relationship Properties](#)

5.6 Moving and Copying Fields

Fields can be moved or copied from one Class to another. Select the field in the source Class and drag it into the target Class to perform a Move. Using the same action with the CTRL Key held down will effect a Copy. To position the transferred

field within a list of existing fields in the target Class, drag the field until the insertion line highlights the desired position.

Moving a Field from a source Class to another Class will have the effect of deleting all data for that particular field in all objects of the source Class!

The reason for this is that Instant Soop is unable to determine the mapping from source objects to target objects - so cannot accurately transfer data during a Field Move. Textual field types may have built up a catalogue of data entries that appear in the drop-down list, and this *is* transferred during a field move, so re-entering data for the field in the target objects may be straightforward.

See Also

[Adding New Fields](#)

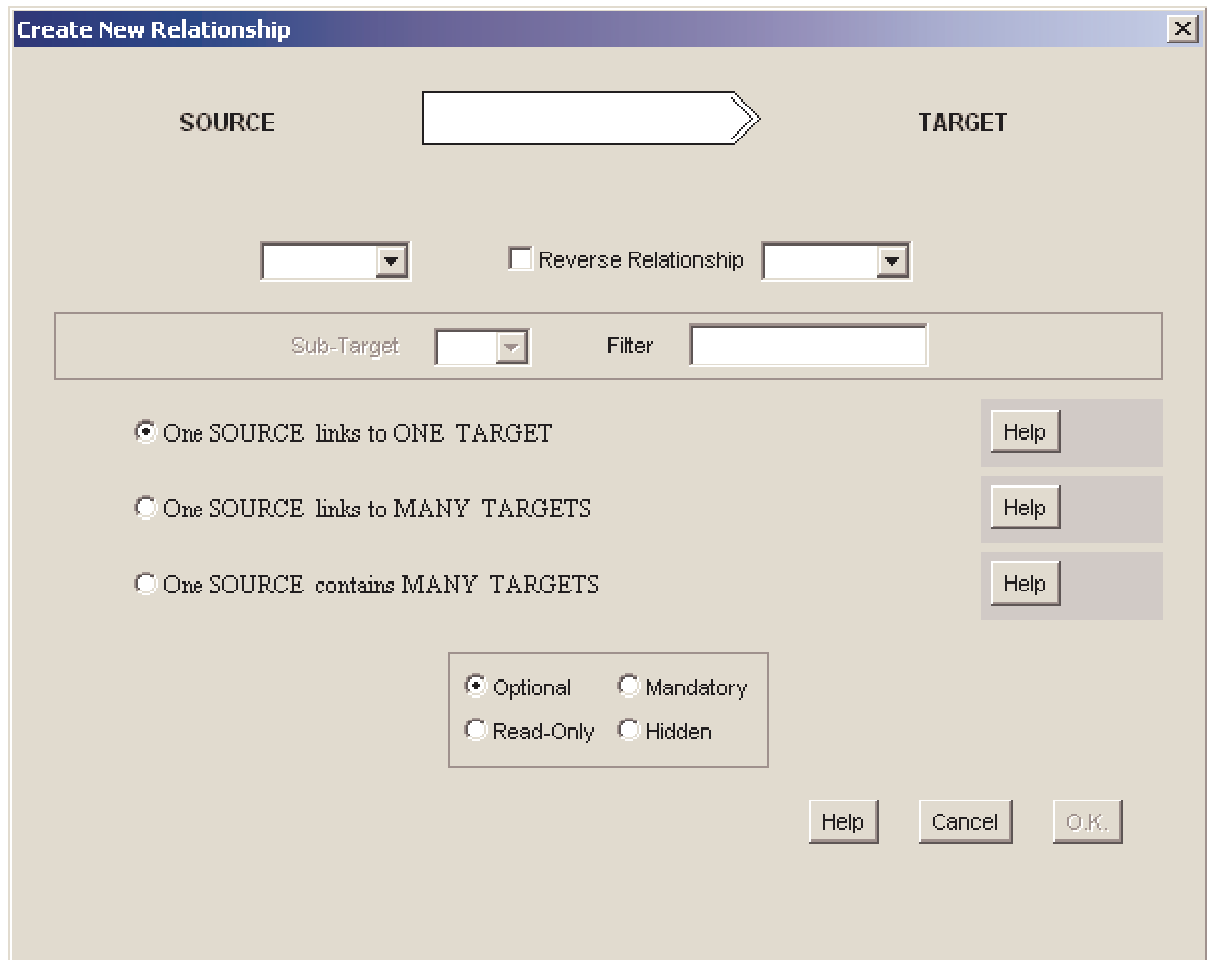
[Moving and Copying Methods](#)

6 Sooplet Relationships

6.1 *Creating a New Relationship*

A Relationship defines how objects of the Source Class relate to objects of the Target Class. Every relationship has a name, a mode and a constraint setting. A Reverse Relationship can also be enabled that links objects of the Target Class back to object(s) of the Source Class.

Choosing 'New Relationship' from the Insert Menu, or 'Add Relationship' from the Class Menu will display the following dialog:



6.1.1 Naming Relationships

The name of the relationship is entered into the arrowed name box. In most cases, a default name will be provided which can be edited if necessary. Names are limited to 16 characters of upper or lower case text.

6.1.2 Choosing the Source and Target Class

The source and target classes are specified by selecting from the drop-down list of classes. If you opened the dialog from the Class Menu, then the source class will already be selected. If you opened the dialog by dragging the Target Class over the Source Class, or by choosing 'Edit Relationship' from the Relationship Menu, then both Source and Target Classes will be pre-set.

6.1.3 Selecting the Mode

The Relationship Mode must be selected from one of the following:

[One-to-One](#) (One Source Object links to One Target Object)

[One-Links-To-Many](#) (One Source Object links to Many Target Objects)

[One-Contains-Many](#) (One Source Object contains Many Target Objects)

6.1.3.1 One-to-One Relationships

One-to-one relationships are the simplest form of relationship, where one source object links to one target object.

For example, in a Personnel Management System, one employee is *linked* to one job.

When the source object is created, it will not be linked to any target object, and may never be linked to a target object, so strictly speaking a one-to-one relationship could be described as one-to-zero-or-one. Read the section on [Constraint Settings](#) to see how such a linkage can be made mandatory.

One-to-one relationships are 'weak' relationships, in that destruction of the source object does not affect the target object, and the same target object can be linked to multiple source objects.

6.1.3.2 One-links-to-Many Relationships

One-links-to-many relationships are the plural form of the one-to-one relationship, where one source object links to many target objects.

For example, in a Sports Club Management System, one team is *linked* to many players.

When the source object is created, it will not be linked to any target objects, and may never be linked to any target objects, so strictly speaking a one-links-to-many relationship could be described as one-links-to-zero-one-or-more. Read the section on [Constraint Settings](#) to see how such a linkage can be made mandatory.

One-links-to-many relationships are 'weak' relationships in that destruction of the source object does not affect the target objects, and the same target objects can be linked to multiple source objects.

6.1.3.3 One-contains-Many Relationships

One-contains-many relationships are the 'strong' form of the one-links-to-many relationship, where one source object contains many target objects.

For example, in an Order Management System, one customer *contains* many orders.

When the source object is created, it will not contain any target objects, and may never contain any target objects, so strictly speaking a one-contains-many relationship

could be described as one-contains-zero-one-or-more. [Constraint Settings](#) are not applicable to this relationship mode, so are disabled.

One-contains-many relationships are 'strong' relationships in that destruction of the source object destroys the contained target objects, and the contained target objects cannot be contained by other source objects.

6.1.4 Constraints Settings

There are 4 options for constraining relationships:

1. [Optional](#)
2. [Mandatory](#)
3. [Read-Only](#)
4. [Hidden](#)

6.1.4.1 Optional Constraint

Optional Relationships, between source and target objects, can be left undefined. This is the position when a new source object is created. If the relationship is a one-to-many, then optional means that the list of target objects can remain empty. This is the default case for contained relationships, that cannot be changed.

6.1.4.2 Mandatory Constraint

Mandatory Relationships, between source and target objects, ensure that a target object must be specified at the earliest opportunity. If the relationship is a one-to-many, then the list of target objects must always contain at least one entry.

6.1.4.3 Read-Only Constraint

Read-Only Relationships, between source and target objects, cannot be edited by users, but are maintained by the Sooplet itself. Reverse Relationships are set to Read-Only and this constraint cannot be removed by the designer.

6.1.4.4 Hidden Constraint

Hidden Relationships, between source and target objects, are not visible so cannot be edited by users. They are available to be manipulated with scripted methods.

6.1.5 Advanced Relationship Options: Reverse Relationships

Relationships are one-way only, linking the source object to the target object. On occasions, an additional synchronized link from the target back to the source is required. As an illustration, in the case of a Customer Object containing a list of Order objects, the Order objects 'know' nothing of the Customer Object in which they are contained. This may be a requirement in an Order Processing System, to be able to locate the customer for an undelivered order.

Reverse Relationships are enabled by ticking the 'Reverse Relationship' checkbox in the Relationship Dialog. A second name box is provided, within the arrow pointing from Target to Source. A default name may be provided, which can be edited if necessary.

6.1.6 Advanced Relationship Options: Sub-Targets

Whilst ordinary relationships link to all the objects in the target Class, Sub-Targets enable the relationship to link to just a sub-set of those objects. When you have a source and a target Class selected, the Sub-Targets drop-down list will be populated with *reachable* collections. Reachable collections may be in the same Class, or a Class that can be reached by navigating through other Relationships. You can select one as the source of your new Relationship. Sub-Targets are only applicable to *One-to-One* and *One-Links-To-Many* Relationship modes.

6.1.7 Advanced Relationship Options: Filters

Filters also enable the list of target objects to be restricted at run-time, but based on their *Field Data* rather than membership of a specific collection. You may require a list of products where the **price > 1.00** or **inStock == Yes**. Just enter the restriction formula into the Filter text box, and when the Relationship is presented to the user, in the form of a drop-down list or Selection View, then the filter will be applied. Filters are only applicable to *One-to-One* and *One-Links-To-Many* Relationship modes.

6.1.8 Editing Relationships

It may be possible to change the mode of a relationship after it has been established, but only in certain cases. The reason for this is that the integrity of data will be compromised once it has been created under certain relationship modes. The following changes are permissible:

One-to-one relationships can be up-sized to one-links-to-many.

One-contains-many relationships can be 'weakened' to one-links-to-many.

For other cases where a mode change is not permitted, the relationship will have to be deleted and recreated. This is a routine operation in Instant Soop.

6.1.9 Renaming Relationships

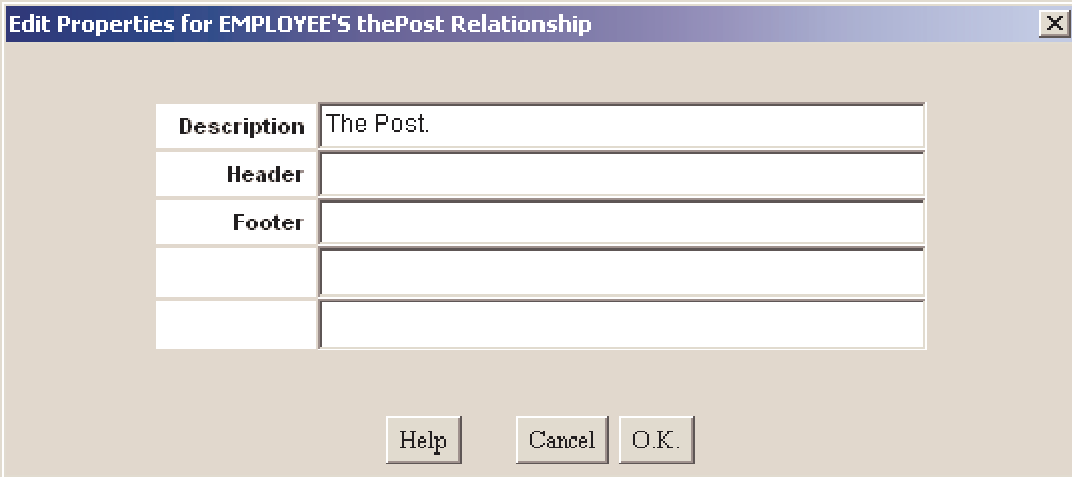
Forward and Reverse Relationships can be renamed by choosing 'Rename Relationship' from the appropriate Relationship Menu. The 'Rename Relationship' Dialog will be displayed to permit this operation.

6.1.10 Deleting Relationships

Forward and Reverse Relationships can be deleted by choosing 'Delete Relationship' from the appropriate Relationship Menu. This action will also delete all data used to represent the linkages between objects enacting this relationship.

6.2 Relationship Properties

The Relationship Properties dialog provides the facility to edit the Description, Header and Footer of a Relationship. The Relationship Properties dialog can be displayed by choosing Edit Properties from the forward or reverse Relationship Menu.



Description	The Post.
Header	
Footer	

Help Cancel O.K.

6.2.1 Description

The Relationship Description is a segment of text that briefly describe the purpose of the Relationship.

The Relationship Description will appear in status text and help text throughout the Sooplet.

6.2.2 Header

The Header is a configurable block that will sit above this relationship in the Data Entry Form. For example, advice for choosing values may be placed in the header.

6.2.3 Footer

The Footer is a configurable block that will sit below this relationship in the Data Entry Form. For example, dividing lines may be placed in the footer.

Headers and footers can contain simple html markup, e.g. `Please choose...` [Please choose...](#)

Headers and Footers are only applicable to One-to-One Relationships in the forward direction.

See Also

[Field Properties](#)

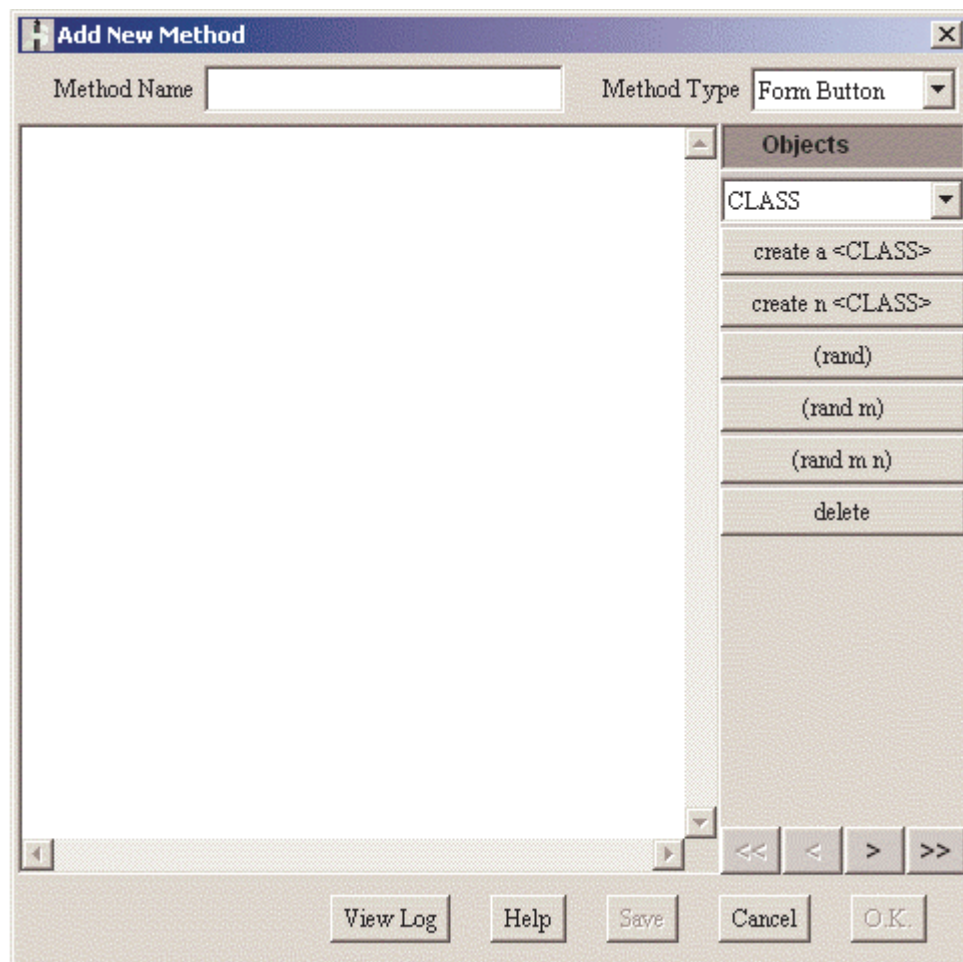
[Class Properties](#)

7 Sooplet Methods

7.1 Creating a New Method

Methods are segments of SoopScript code that define some behaviour for the owning Class. Method names are restricted to 16 characters in length, and can comprise upper or lower case characters, and digits. Spaces are not permitted in Method names. Some Method names may be rejected due to clashes with Instant Soop reserved words.

Choosing 'Add Method' from the Class Menu will display the following dialog:



The **Method Name** is entered in the 'Method Name' box, and can also be changed here subsequently. The **Method Type** is selected from the drop-down list. When certain

method types are selected, the method name will be set for you, and cannot be changed. A useful comment will be generated as the first line for each method. This comment can be edited to make it more useful, as it appears in the help system and the status bars of the Form and Data Entry Views.

There are eight types of method available: Table Button, Form Button, Object Creation, Data Edit, Data OK, Private, Hidden and Pulse. Each of these types is described below. To the left of the dialog is the script pane where SoopScript is typed. To the right of the script pane is the SoopScript Toolbar, which provides a selection of command buttons and drop-down lists to assist with the entry of SoopScript. Methods can be saved using the Save Button, which will keep the Method Editor Dialog open for re-editing. Clicking OK will save the method and close the dialog. Click the 'View Log' button to view a log of the actions recorded when the method was last executed. Click the 'Edit Method' button that is now visible to return to the script pane.

7.1.1 Table and Form Button Methods

Button Methods are exposed as buttons on the Sooplet Data Entry View and Form View for Objects of the owning Class. The Method is executed when this button is clicked.

7.1.2 Object Creation Method

Object Creation Methods are executed when an Object of the owning Class is created. This action is suppressed when objects are imported via a script.

7.1.3 Data Edit Method

The Data Edit Method is executed when the Form for a particular Object is opened. This provides an opportunity to capture data before it is edited, to configure button states or to perform other tasks.

7.1.4 Data OK Method

The Data OK Method is executed after the OK Button on a Form is clicked. This provides an opportunity to access data after it has been edited.

7.1.5 Private Method

A Private Method will not appear on any user-accessible buttons, but can be useful for holding library routines that can be called from other methods.

7.1.6 Hidden Method

A further method type 'Hidden' is similar to Private, but Hidden methods are not visible in the Class Diagram. They cannot be edited as other method types, and can only be created using the method creation command within another method. See help on the SoopScript Command Language for further details.

7.1.7 Pulse Method

The Pulse method type gives objects a regular heartbeat, to allow them to perform routine operations on themselves. The default period for the pulse method to run is every 15 seconds, but this can be re-defined in the [Sooplet Properties](#) file.

7.2 Deleting a Method

Choosing 'Delete Method' from the Method Menu will delete the method.

7.3 Renaming a Method

Choose 'Edit Method' from the Method Menu to display the Edit method dialog which enables the Method name to be changed.

7.4 Editing a Method

To edit a Method after it has been created, choose 'Edit Method' from the Method Menu or double-click the Method name in the Class Diagram.

7.5 Splitting a Command over multiple Lines

To make a Method more readable, commands can be split over multiple lines with the command continuation character _ (underscore). This character must be placed at the end of a line, where a space may be expected.

See Also

[SoopScript Command Language](#)

7.6 Moving and Copying Methods

Methods can be moved or copied from one Class to another. Select the method in the source Class and drag it into the target Class to perform a Move. Using the same action with the CTRL Key held down will effect a Copy. To position the transferred method within a list of existing methods in the target Class, drag the method until the insertion line highlights the desired position.

A Method that works in one Class may fail when copied to another Class! Some editing may be necessary to adjust the script to its new Class.

See Also

[Moving and Copying Fields](#)

8 Sooplet Data

8.1 Delete All Data

Choosing 'Delete All Data' from the Sooplet Menu will delete all the objects, collections and data in your Sooplet, but not the classes that make up the design. Be sure this is what you want to do, before answering 'Yes' to the warning prompt.

See Also

[Delete All Classes](#)

9 Context Menus

In addition to the Instant Soop Menu Bar, visible above the Sooplet Diagram, there are a number of Context Menus that are tailored to particular features of the Sooplet design. Context Menus are displayed by Right-Clicking the relevant feature of the Sooplet Diagram.

Some operations are only available through Context Menus.

1. [The Sooplet Design Menu](#)
2. [The Class Menu](#)
3. [The Field Menu](#)
4. [The Method Menu](#)
5. [The Relationship Menu](#)
6. [The Reverse Relationship Menu](#)

9.1 The Sooplet Design Menu

This menu is accessed by right-clicking a blank area of the Sooplet Diagram and has the following options:

- Add Class
- Delete All Classes

9.2 The Class Menu

This menu is accessed by right-clicking a Class Heading and has the following options:

- Add Field
- Add Method
- Add Relationship
- Add Child Class
- Set Badge
- Delete Class
- Edit Properties
- Edit Permissions

9.3 The Field Menu

This menu is accessed by right-clicking a particular Field within a Class and has the following options:

- Delete Field
- Rename Field
- Set Data Type
- Set Initializer
- Edit Properties

9.4 The Method Menu

This menu is accessed by right-clicking a particular Method within a Class and has the following options:

- Delete Method
- Edit Method

9.5 The Relationship Menu

This menu is accessed by right-clicking the name of a particular Relationship. This will be the upper name in a Relationship with forward and reverse components. The menu has the following options:

- Delete Relationship
- Rename Relationship
- Edit Relationship
- Edit Properties

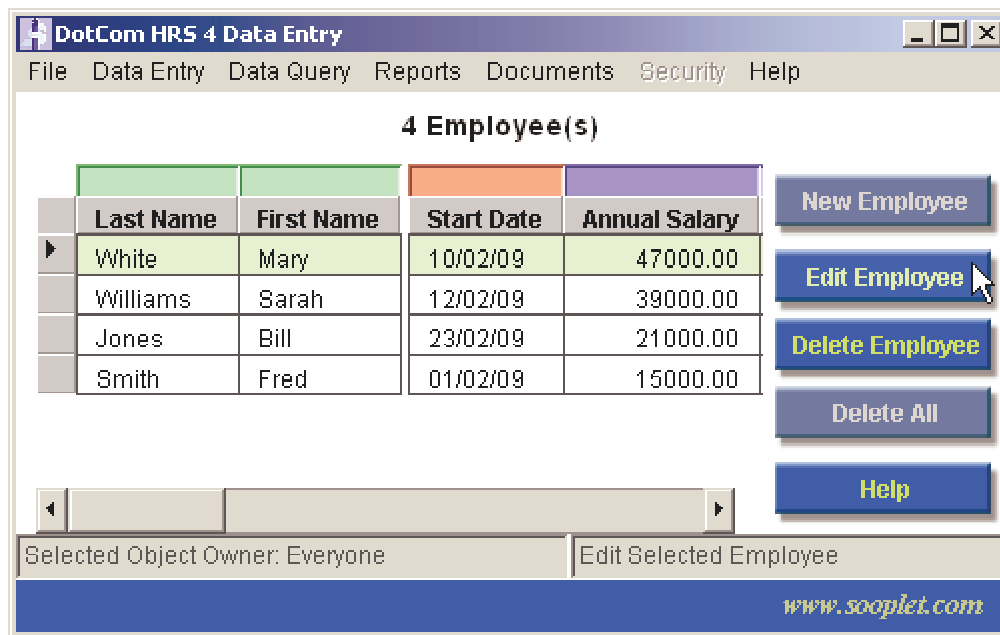
9.6 The Reverse Relationship Menu

This menu is accessed by right-clicking the name of a particular Reverse Relationship and has the following options:

- Delete Relationship
- Rename Relationship
- Edit Properties

10 Preview Sooplet

The 'Preview Sooplet' function opens the current Sooplet in a separate window. If the Sooplet is already opened then it will be brought into view.



The current Sooplet can be closed at any time to enable the designer to focus on the design, before previewing the Sooplet to confirm the design is implemented as expected.

Opening a Sooplet in preview mode is optimised for efficiency, so there are some differences to running a Sooplet in standalone mode, from say a desktop shortcut. For example, the splash screen is disabled.

11 Creating a Desktop Shortcut

A shortcut to the Sooplet under construction can be placed on 'All Users' desktops and in their Start menus by choosing the 'Create Shortcuts' option from the Application menu. Should your platform or environment prevent the automatic creation of a shortcut, then you can set one up manually.

11.1 Setting a Shortcut Manually (Windows)

1. Shortcuts point to the Instant Soop Executable (iSoop.exe), and pass the name of the Sooplet as a parameter
2. Locate the executable in the Instant Soop installation folder ("C:\Program Files\Warrensoft\Instant Soop 4.1\iSoop.exe" in the standard installation).
3. Right click it, and choose create shortcut.
4. Add a parameter for the default SoopFiles location, usually "all"
5. Add a parameter for the Sooplet Name - this may require double quotes
6. Move the shortcut to your desktop, start menu folder or other location.

Example "C:\Program Files\Warrensoft\Instant Soop 4.1\iSoop.exe" all "DotCom HRS"

11.2 Setting a Shortcut Manually (Java Webstart for Windows)

1. Shortcuts point to a Java Network Launch Protocol (JNLP) descriptor which is created for the Sooplet, in the same way that Instant Soop is run under Java Webstart
2. Locate the <SoopletName>.jnlp file in the Instant Soop installation folder (%ALLUSERSPROFILE%\Application Data\Warrensoft\InstSoop4\SoopFiles in the standard installation).
3. Right click it, and choose create shortcut.
4. Move the shortcut to your desktop, start menu folder or other location.

Example "C:\Documents and Settings\All Users\Application Data\Warrensoft\InstSoop4\SoopFiles\estab.jnlp"

11.3 Setting a Shortcut Manually (Java Webstart for Linux)

1. Shortcuts point to a Java Network Launch Protocol (JNLP) descriptor which is created for the Sooplet, in the same way that Instant Soop is run under Java Webstart
2. Locate the <SoopletName>.jnlp file in the Instant Soop installation folder (%HOME%/Warrensoft/InstSoop4/SoopFiles in the standard installation).
3. Copy the jnlp file to your desktop, menu or other location.

Example "file:/home/user/Warrensoft/InstSoop/SoopFiles/estab.jnlp"

See Also

[Deploying a Sooplet](#) for more information about packaging your Sooplets for sale or public distribution.

12 Deploying a Sooplet

There are various options for deploying Sooplets:

[Deploying Sooplet as a remote Java Webstart Application](#)

[Deploying Sooplet as a local Java Webstart Application](#)

[Deploying Sooplet to Sooplet Server](#)

12.1 Deploying Sooplet as a remote Java Webstart Application

Sooplets can be deployed to remote websites as Java Webstart Applications, in the same way that Instant Soop is made available from the Sooplet.com website. Choose 'Deploy Sooplet' from the Sooplet Menu, fill in the necessary details and click Publish. By default, the webpage will be named the same as your Sooplet, however this can be changed by [setting the 'deployHomepage' property](#).

To provide a 'one-click' installation experience for users of your Sooplet, the hosting webserver should be configured to support the jnlp mime type. One method of achieving this is to [set the jnlpMimeHeader Instant Soop property](#). See the java webstart documentation at www.java.sun.com for further information.

Sooplet.com provides a free facility for you to deploy, test and promote your Sooplet - called *The Gallery*. The default ftp settings for deploying a new Sooplet will be set to point to *The Gallery* [Server ([ftp.sooplet.com](ftp://ftp.sooplet.com)), directory (/), username (gallery@sooplet.com), password (thankyou)].

12.2 Deploying Sooplet as a local Java Webstart Application

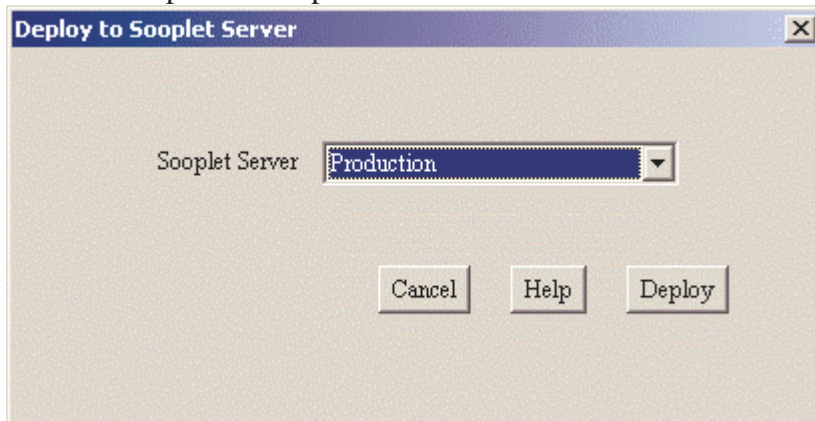
The above technique can be used to deploy Sooplets to an Intranet or local network webserver. Alternatively, the 3 files required (webpage, jnlp descriptor, jar file) will be found in the deploy subfolder of the default Instant Soop installation folder, following a deploy operation, should you wish to deploy by other means.

12.3 Deploying Sooplet to a Sooplet Server

Sooplet Server is a server-based product for hosting Sooplets. Sooplets are then available to a web browser, or downloadable client application. By default, Instant Soop uses a different folder location to the Sooplet Server, so as to keep development and production Sooplets apart when developing and testing on a single workstation. The Sooplet Server can be configured to use the same folder as Instant Soop, but this is not recommended. It is advisable to keep development and production folders separate, and use one of the deployment techniques described below. Deploying to the Sooplet Server can be done in three ways: Auto-Deploy, FTP Publishing or Manual Copy.

12.3.1 Auto-Deploy

The auto-deploy function will seek out Sooplet Servers on your network and provide a drop-down list from which to choose the deploy target. To auto-deploy choose 'Deploy to Sooplet Server' from the Sooplet menu, select a target server and click Deploy. Your packaged Sooplet will be moved to the Sooplet Server, unpacked and included in the directory at Sooplet Central. The distribution file will be renamed (.000, .001, etc.) should the need arise to roll back. The Sooplet Server will recognize the new or updated Sooplet without the need to restart.



If your network prevents the auto-discovery of Sooplet Servers, or you require more control over the deployment then FTP Publishing will be the best method.

12.3.2 FTP Publishing

To be able to deploy a Sooplet to the Sooplet Server using FTP Publishing, the server must have a FTP server installed, with access to the SoopFiles folder. If this is the case, then follow the instructions for 'Deploying Sooplet as a remote Java Webstart Application' above. Complete the server name, folder location, username and password and click Publish. Your packaged Sooplet will be moved to the Sooplet Server, unpacked and included in the directory at Sooplet Central. The distribution file will be renamed (.000, .001, etc.) should the need arise to roll back.

12.3.3 Manual Copy

This method involves copying the set of files named after the sooplet (DotCom.spi, Dotcom.sdt, etc.) to the Sooplet Server's SoopFiles folder. The Sooplet will be discovered by the Sooplet Server and appear in the directory at Sooplet Central.

See Also

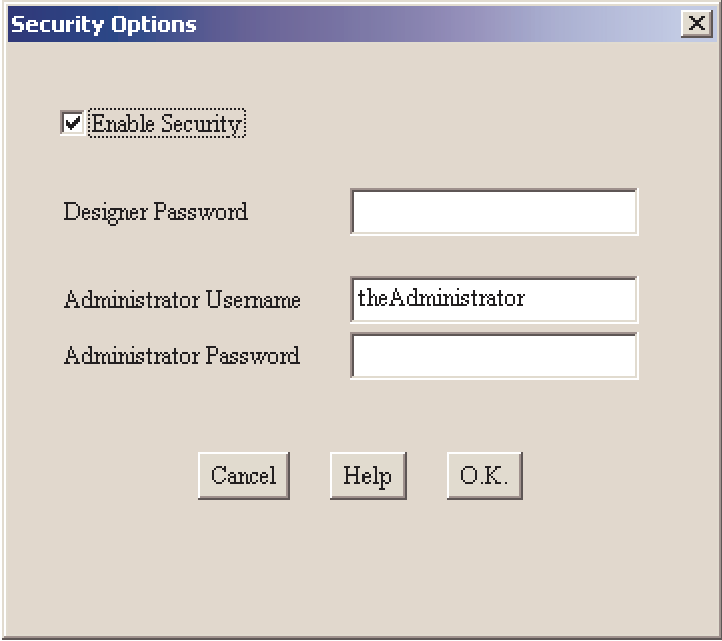
[Configuring Instant Soop Properties](#)

[Configuring Sooplet Properties](#)

13 Sooplet Security Options

Sooplets can be secured to ensure that only users with the requisite permissions are able to perform certain functions. Roles and Permits can be created at design time by the designer, or at run time by the Sooplet Administrator. The task of creating users and assigning roles in a secure Sooplet, falls to the Sooplet Administrator. The purpose of the Security Options dialog in Instant Soop is to secure or unsecure a Sooplet, accept the username and password to empower the Sooplet Administrator, and accept a password to subsequently protect the Sooplet design.

The Security options can be accessed by choosing Security from the Tools menu. The following Dialog will be displayed:



The screenshot shows a dialog box titled "Security Options". It features a checked checkbox labeled "Enable Security". Below this are three text input fields: "Designer Password" (empty), "Administrator Username" (containing the text "theAdministrator"), and "Administrator Password" (empty). At the bottom of the dialog are three buttons: "Cancel", "Help", and "O.K."

13.1 Enabling Security

Ticking the 'Enable Security' checkbox will turn the Sooplet into a secure Sooplet. All users will need to logon to access the Sooplet, although network usernames can be used to effect a 'single sign-on'.

Enter details for the Administrator's password and Username (or accept the default).

It is important to remember that Securing a Sooplet will restrict access to authorised users only. The Administrator account is critical to managing a Secure Sooplet. Be sure to take steps to safeguard this information.

Provide a Designer Password to protect the design and the security options that have been configured.

No access can be gained to a Secured Sooplet Design without the Designer password. When this Sooplet is subsequently opened in Instant Soop, then the password will be requested, and failure to supply the correct password will deny access to the design.

See Also

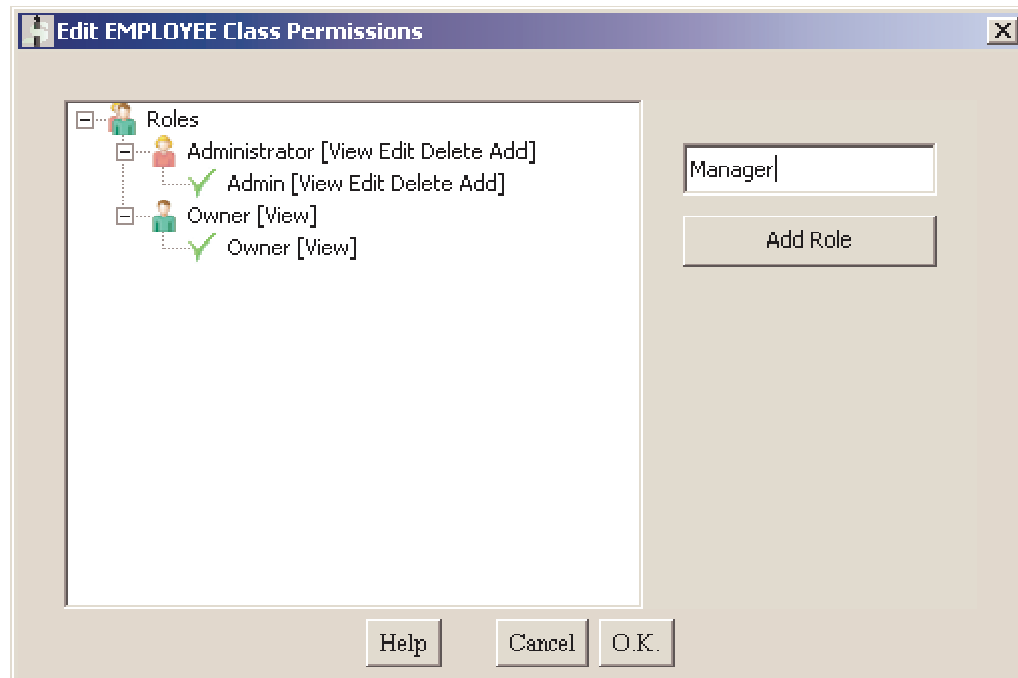
[Class Permissions](#)

[Secure Sooplets](#) in the Sooplet User Guide.

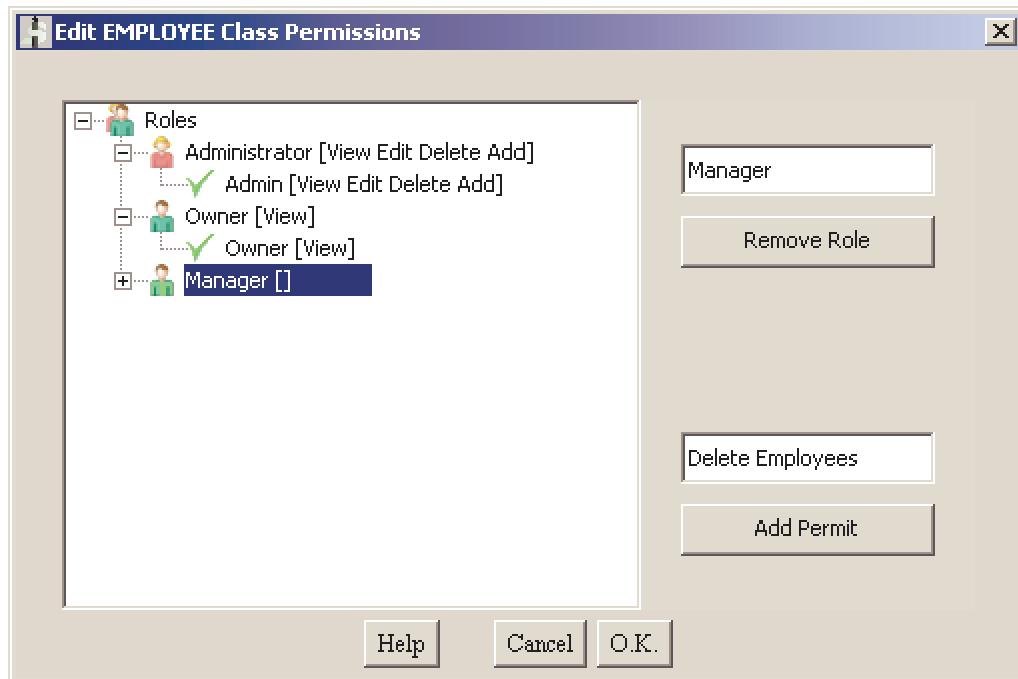
13.2 Class Permissions

The Class Permissions dialog provides the facility to configure the roles and permits that control what actions can be performed on a particular class's objects. These permissions are only applied in secure Sooplets, but can be set up at any time by the designer. For example, you may want only managers to be able to delete EMPLOYEE objects. This can be achieved in the Instant Soop designer by creating the 'Manager' Role, adding the 'Delete Employees' Permit, and then within the secure Sooplet assigning a User to the Role of 'Manager'.

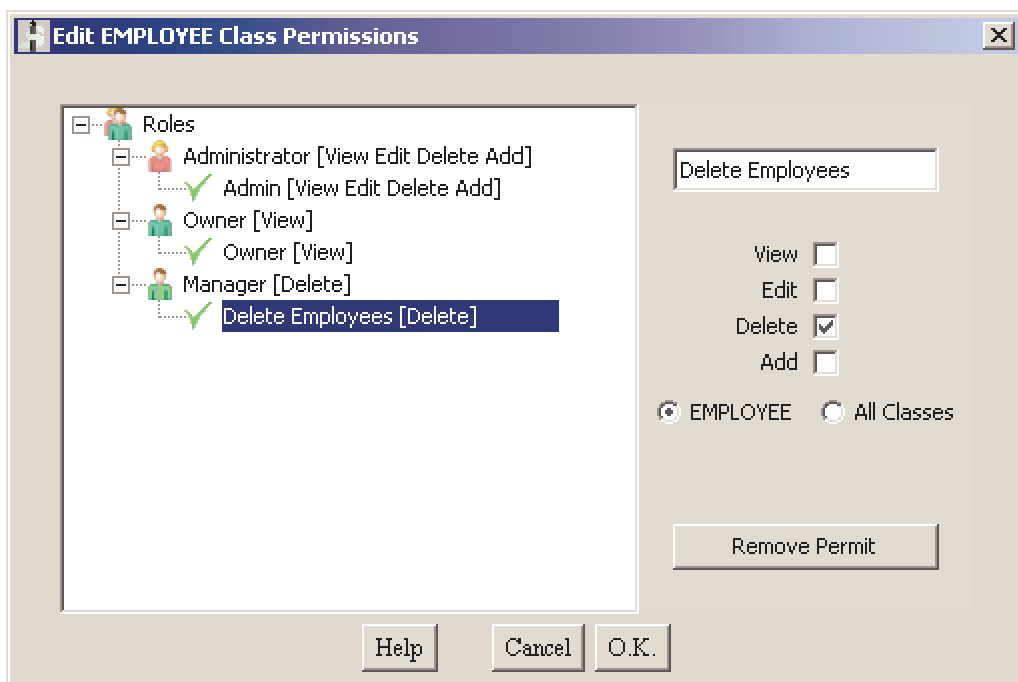
The Class Properties dialog can be displayed by right-clicking a class and choosing Edit Permissions from the Class pop-up menu.



This dialog enables roles to be added, edited or removed. When a new role is added, a dialog box appears enabling a new permit to be named.



The final step is to define the operations that the permit allows. The allowed operations are listed against the individual permit in the left-hand pane, and aggregated to show the full permissions for the role.



See Also

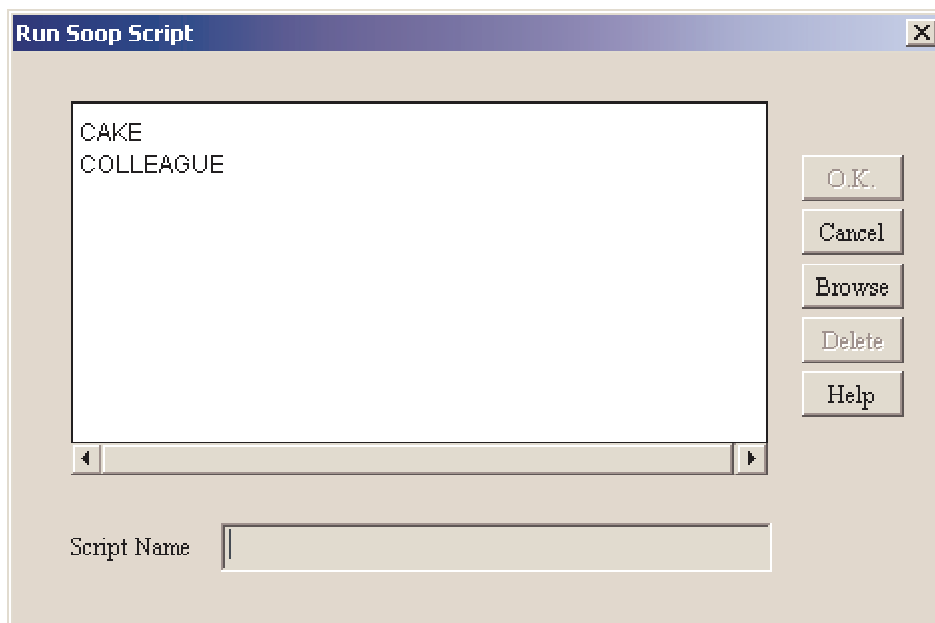
[Sooplet Security](#)

14 Soop Scripts

Soop Scripts enable the Instant Soop designer to automate a number of tasks in a consistent, repeatable fashion. Single classes can be saved as scripts and imported into other Sooplets, building a library of re-usable components. Entire Sooplets can be saved in a single script, for back-up and maintenance purposes.

14.1 Running a Script

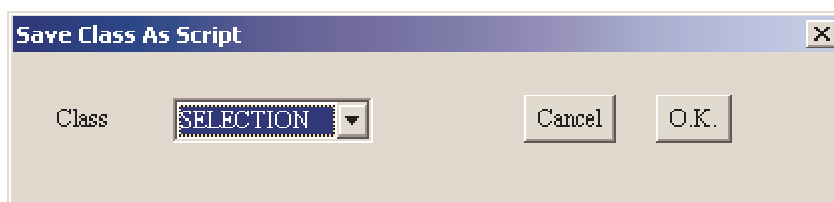
Choose Run from the Script Menu and the following dialog will be displayed.



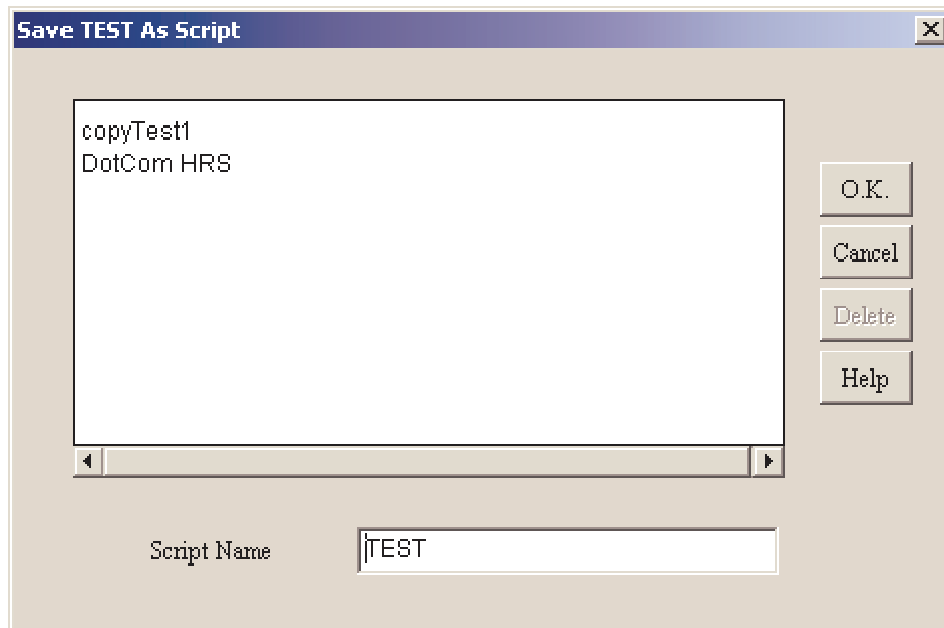
The centre pane of the dialog provides a list of current scripts. Run a script by double-clicking it in the list, or clicking once to select it and then clicking the OK Button. Selected scripts can also be deleted from this dialog. If a script has been composed outside of Instant Soop, or downloaded from the web, then the Browse facility can be employed to locate the script in order to run it.

14.2 Save Class As Script

Choose 'Save Class' from the Script Menu and the following dialog will be displayed:



Select the class you want to save from the drop-down list, click OK and the following dialog will be presented:



By default the class name will be used as the script name, but you can change this by editing the name in the Script Name box, or by clicking an existing script name in the list and editing it. When you have completed the script name click OK to save the script. Selected scripts can also be deleted from this dialog.

14.3 Save Sooplet As Script

Choose 'Save Sooplet' from the Script Menu to save the current Sooplet as a script. The Save As Script dialog will be displayed, as with saving classes as scripts, and the script name will be based on the Sooplet name.

See Also

[Soop Script Command Library](#)

15 Printing a Sooplet

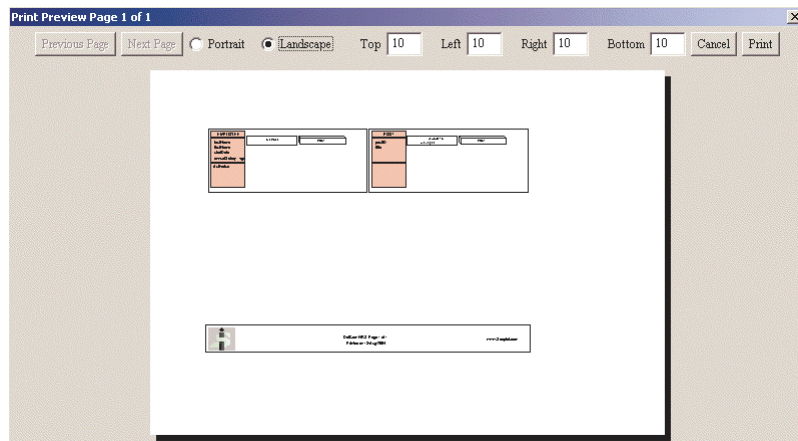
Choosing Print from the File Menu will print the current Sooplet Design on the default printer.

See Also

[Print Preview](#)

15.1 Print Preview

Choosing Print Preview from the File Menu will display the Print Preview Dialog.



The dialog enables the orientation and margins to be set, and the pages to be previewed before printing.

See Also

[Printing a Sooplet](#)

16 Configuring Instant Soop Properties

There are various options for configuring the behaviour of Instant Soop. Settings in the 'InstSoop.dat' file, which is located in the default installation folder, can be edited with a plain text editor such as Notepad. The property value following the = sign can be changed, before saving the file. The properties and default settings which are user-configurable are described in detail under the following categories:

16.1 RTF Document Settings

The following 6 properties are used in the layout of Documents:

```
DefaultPixelsPerChar=6.2
DefaultPixelsPerTab=36
DefaultWPFontName=Serif
DefaultWPFontSize=12
DefaultInterTextCRCount=1
DefaultCharLimitBeforeCR=60
```

The following properties enable the path to the RTF compatible Word Processor to be specified. The Sooplet will try each entry until it succeeds.

```
RTFApplicationPath1="C:\\Program\ Files\\Microsoft\
Office\\Office\\WinWord.exe"
RTFApplicationPath2="D:\\Program\ Files\\Microsoft\
Office\\Office\\WinWord.exe"
```

16.2 Most Recently Used File Lists

The number of entries in the most recently used lists can be specified:

scriptMRUMax=10

fileMRUMax=10

16.3 Scripts

When data is imported via a script, objects owned by unknown owners will be assigned to the administrator if the value of this property is admin. Otherwise a new user will be created, with a random password.

UnknownOwnerAction=admin

On-Create Methods (constructors) are normally executed when an object is created. This action may not be desirable when objects are imported via a script. The InhibitConstructor property enables script driven constructors to be inhibited, however, the default (no) is to allow them.

InhibitConstructor=no

16.4 Sooplet Help

The default port number for serving up Sooplet Help Pages can be specified. This may need changing to avoid any clashes with existing services.

WebServerPort=6788

16.5 Regional Settings

The default locale for Instant Soop can be specified using language and country. This impacts the way dates are presented:

language=en (2 character ISO 639 code e.g. ar [Arabic], de [German], es [Spanish], etc.)

country=GB (2 character ISO 3166 code e.g. US [United States], FR [France], etc.)

16.6 Deployment Settings

To provide a 'one-click' installation experience for users of your Sooplet, the hosting webserver should be configured to support the jnlp mime type. If this is not possible, then an alternative method is to place an extra line in the jnlp file to inform the webserver it is a jnlp file.

```
jnlpMimeHeader=<? header ("Content-Type\: application/x-java-jnlp-file"); ?>
```

To remove this line from the jnlp file, change the property to `jnlpMimeHeader=`

16.7 EMail Settings

The Simple Mail Transfer Protocol (SMTP) used by the `sendmail` command requires the URL of the mail server providing the SMTP service. This information will be provided by your Internet Service Provider or Systems Administrator. An example entry would look similar to this:

```
smtpHost=smtphost.mailserver.com
```

16.8 Pulse Period

Sets the repeat period in seconds for executing Pulse methods.

```
pulsePeriod=15
```

See Also

[Configuring Sooplet Properties](#)

17 Configuring Sooplet Properties

There are various advanced options for configuring the behaviour of Sooplets. Settings in the '<soopletname>.dat' file, which is located in the `SoopFiles` subfolder of the default installation folder, can be edited with a plain text editor such as Notepad. The property value following the `=` sign can be changed, before saving the file. The properties and default settings which are user-configurable are described in detail under the following categories:

17.1 Form Settings

The following properties are used in the layout of Forms:

`MaxOneLineTextFieldSize=40` sets the point at which a line of text becomes a scrolling text area

`CheckBoxObjectLimit=3` sets the point at which a radio selection becomes a drop-down list

17.2 Reports

17.2.1 Prefixes and Suffixes

`includePrefixesInReportData=default`
`includeSuffixesInReportData=default`

The default behaviour for prefixes and suffixes in data-oriented reports (xml and csv) is for them to be included in badge columns but not in the body of an object. Change the setting to 'always', to have prefixes or suffixes always included, or 'never' to exclude them from all columns of data-oriented reports.

17.2.2 Relationship Names

`incRelationshipNamesInXMLReport=default`

By default, relationship names are included in XML Relation reports but not in XML Routed reports. Change the setting to 'always', to have relationship names included in either, or 'never' to exclude them from xml reports completely.

17.3 Deployment

The page name for the Sooplet deployment website can be specified:

`deployHomepage=home.htm`

If the value is left blank (the default), then the page will be named the same as the Sooplet. Use this setting to publish your Sooplet to the free test hosting facility - the Gallery.

17.4 Object Locking

Object Locking is important when you have multiple users accessing the same Sooplet, hosted by the Sooplet Server. Locking will be consistently applied in both the Dedicated Client and the Browser Client. There are 3 options for specifying the way Sooplets lock objects that are being edited: Pessimistic, Optimistic and No Locking.

`lockMode=pessimistic`

Pessimistic locking is the most protective setting. If a user opens the Form View to edit an object, then no other user will be able to open the Form View on that object, until the first user has closed the Form View. Pessimistic locking requires users to observe certain rules to ensure they do not lock objects unnecessarily, or for extended periods. Good practice is to:

- Always log off from a session when you have finished using it
- Don't leave the Form View open unnecessarily

If an object is pessimistically locked when you try to edit it, then a warning message will be displayed informing you of the user and machine holding the lock. For this reason pessimistic locking is not enforced in an unsecured Sooplet, and the locking mode reverts to Optimistic.

lockMode=optimistic

Optimistic locking is the default for new Sooplets. Multiple users can edit the same object, but when submitting changes the user will be notified if the object has been changed in the meantime. The latest changes would be discarded in these circumstances. Optimistic locking requires less discipline from users as they do not hold persistent locks on objects.

lockMode=none

Setting the lock mode to 'none' turns off locking completely. Multiple users can edit the same object, with the latest user's changes persisting. Even with the lock mode set to none, multiple method requests are protected from being run on the same object at the same time.

17.5 Pulse Period

The repeat period for pulse methods can be specified. The value is in seconds:

pulsePeriod=15

18 Closing a Sooplet

To close a Sooplet choose Close from the File Menu. The diagram will be removed ready for a new Sooplet design to be created.

See Also

[Opening a Sooplet](#)

19 Exiting Instant Soop

Choose Exit from the File Menu to close the Instant Soop designer application and any Sooplet under construction. The last Sooplet will be automatically recalled when the Instant Soop designer is re-opened. In addition, a list of the most recently used Sooplets appears at the foot of the File menu.

There is no 'Save' option on the File Menu as any changes to the Sooplet design are automatically recorded.

See Also

[Opening a Sooplet](#)

20 Help

20.1 Instant Soop Help Index

20.1.1 Menubar Menu Options

File	Insert	View	Script	Sooplet	Tools	Help
New	New Class	Association	Run	Preview Sooplet	Security	Help Menu
Open	Class From Database	Inheritance	Save Class	Create Shortcuts		
Close	New Relationship		Save Sooplet	Deploy Sooplet		
Print Preview				Deploy to Sooplet Server		
Print				Delete All Classes		
Exit				Delete All Data		

20.1.2 PopUp Menu Options

Design Menu	Class Menu	Field Menu	Method Menu	Relationship Menu
Add Class	Add Field	Delete Field	Delete Method	Delete Relationship
Delete All Classes	Add Method	Rename Field	Edit Method	Rename Relationship
	Add Relationship	Set Data Type		Edit Relationship
	Add Child Class	Set Initializer		Edit Properties
	Set Badge	Edit Properties		
	Delete Class			

	Edit Properties			
	Edit Permissions			

20.2 GLOSSARY

Class

An entity that represents a significant element of the application area which the [Sooplet](#) is designed to manage. For example a CUSTOMER, ORDER or PRODUCT. Classes are singularly named, and hold the specification for [Fields](#), [Methods](#) and [Relationships](#).

Field

A slot within a [Class](#) to hold data about an [Object](#).

Object

An entity that holds the data for a specific element within the application area, such as a Customer Object. A [Sooplet](#) will contain many objects, each one dynamically referencing its [Class](#) to understand what data it should hold and how it should behave.

Collection

A batch of objects, usually of the same [Class](#).

Method

A segment of [SoopScript](#) that defines some behaviour or calculation relevant to that class of [Object](#).

Script

A segment of [SoopScript](#) that can be used to create and manipulate [Sooplet](#) designs.

Class Heading

The area of the Class representation on the Sooplet Diagram that holds the name of the [Class](#).

Sooplet

A computer application that manages information for a particular problem domain.

Instant Soop

An application for rapidly designing and deploying [Sooplets](#) without writing code.

Sooplet Diagram

The drawing canvas where elements of the [Sooplet](#) design are displayed.

Relationship

The connection between two [Classes](#), the Source and the Target, or between a [Class](#) and itself.

Reverse Relationship

The connection between a Target [Class](#) and its Source [Class](#).

Polymorphic Relationship

A [Relationship](#) with a Class and its child classes.

SoopScript

A script language specifically designed for the manipulation of [Classes](#) and [Objects](#).

Child Aggregation

A technique for combining a list of data into a single cell, useful in [CSV reports](#) and [Queries](#) that form the datasource for [Sooplet Documents](#).

CSV

Comma Separated Values - a file format used to represent a block of data in a table-like way.

XML

Extensible Mark-up Language - a file format used to represent data in a tagged tree-like structure.

HTML

Hypertext Mark-up Language - a file format used to present data in a Browser.

TXT

Text - a simple format that can be viewed in any Text Editor e.g. Notepad.

Sooplet Table

The main view of a collection of Sooplet objects, reached from the data entry menu .

Polymorphic Table

A [Sooplet Table](#) that shows objects from a Class and its child classes.

Query

A technique for extracting a subset of the required data from a Sooplet. A query forms the basis of a [Sooplet Document](#).

Sooplet Document

A word-processed file that forms the template for a business publication such as an Invoice, Letter or Delivery Note.

Report

A file, created automatically by the Sooplet, representing selected information in a range of formats.

User

A person with a login to a Secure Sooplet

Role

A set of responsibilities that a User has, which require special permissions to manipulate Sooplet data.

UserRole

A Sooplet Role that a User can assume.

Permit

The right to Add, Edit, View and Delete objects of a particular Class.

OwnerPermit

The right to Edit, View and Delete objects owned by the User.

^CLASS

The Class, plus all classes that inherit from that class.