

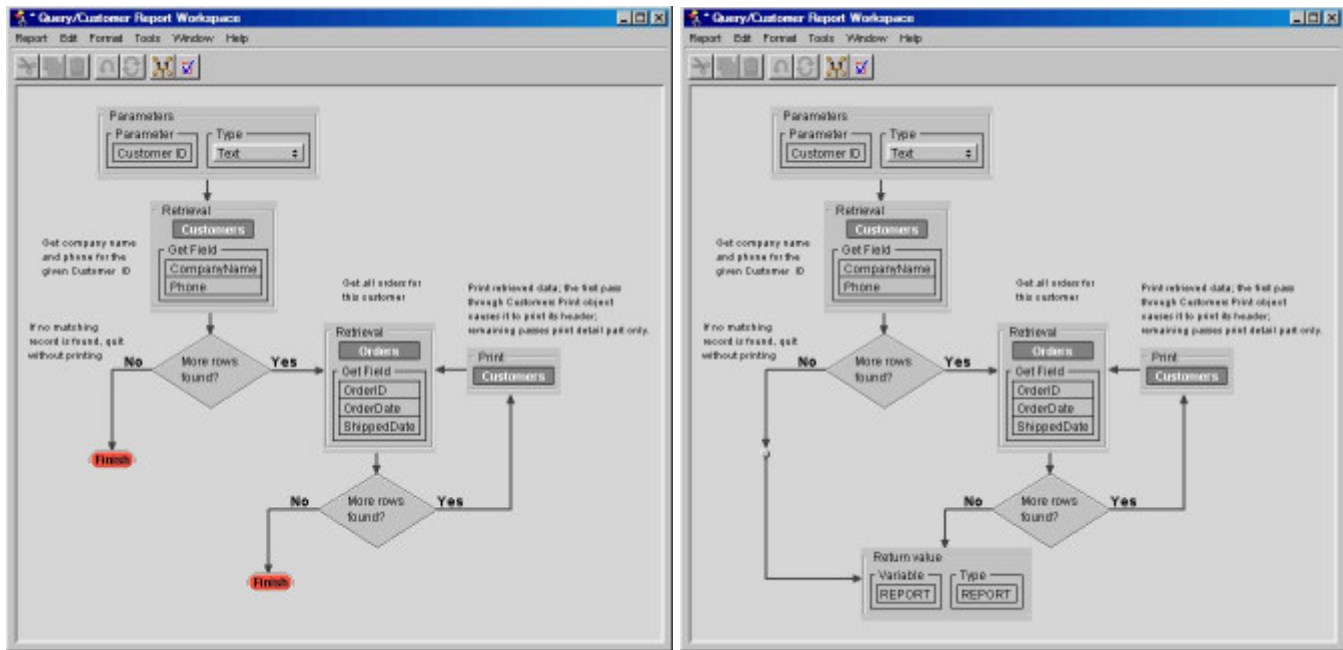
Tutorial 13 - Sending reports via e-mail

The standard way of producing reports in Scribe is by displaying them in a dedicated window, or saving them in the database for subsequent display or printing. The reports are produced as collections of Java objects and can be either displayed by Scribe, or by another Java application that makes use of Scribe API.

Alternatively, Scribe can be instructed to create reports in the HTML or PDF format. The formatted report is stored in volatile memory and can be passed to some other function for further processing (e.g. the e-mail function that can send the report as a PDF attachment, or an HTML-based message, to the specified address).

In this Tutorial we will create one such report, and learn how it can be e-mailed.

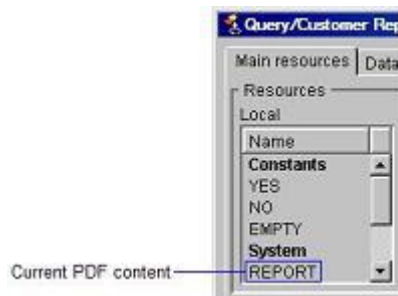
We will reuse the Customer report created in the Tutorial 4, and replace the Finish components with the Return:



Original procedure (with Finish)

Altered procedure (with Return)

The Return component in the altered Customer report is set to a system variable REPORT that is provided by the Palette:



The REPORT variable is set by Scribe to the formatted copy of the currently produced report. It should be used at the end of the Report Procedure, after executing all Print components. The reason for this requirement is that the content of the REPORT variable is set only once per procedure, and it happens when the variable is referenced the first time.

We drag the REPORT variable into the Return component; when the report executes, its formatted content is

placed into this variable just before the Report Procedure exits. Of course, there is not much use for the REPORT variable in the new version of the Customer report, since it happens just before the completion of the procedure, but if the report is used as a "black box" in some other procedure, the returned variable with formatted content can be used in e.g. passing the PDF attachment to an e-mail function. We will look at one such procedure in the next section.

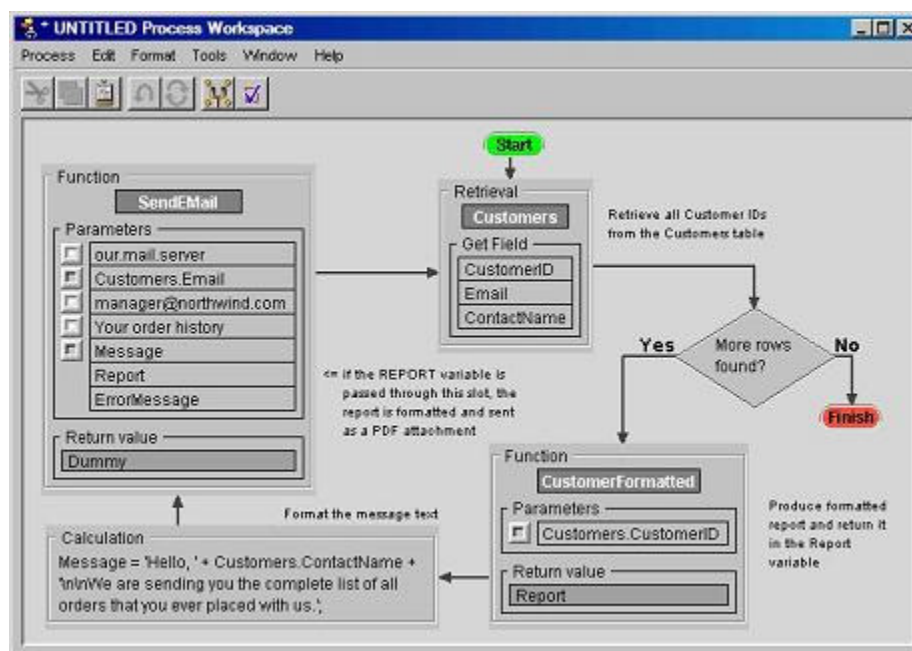
Save the Report Procedure as Query/CustomerFormatted.

Scribe Process Procedure

Process Procedures in Scribe are flowcharts like User Functions and Reports. The only difference is that the Process Procedures have access to a somewhat different set of executable components, with components specific to the Reports removed, and components for database table updates (Insert, Update, Delete and Transaction Packet) added.

The Process Procedure that we will create in this section does not make use of the extended database handling capabilities of Processes, and strictly speaking could be replaced by an equivalent Report Procedure, but we will use it just to introduce the new Scribe feature.

Select Process Procedures entry in the Application Launchpad window and click on the Launch button (or just double-click the Process Procedures entry). Scribe will open a new ("UNTITLED") Process Procedure Workspace window, and we will assemble the following flowchart in it:



This flowchart runs a data retrieval loop; the Retrieval component extracts all Customer IDs, their e-mail addresses and contact names from the Customers table, sorted by Customer ID (set in the Inspector not shown here), and passes every returned Customer ID to the Customer report. The Customer Report Procedure is encapsulated in the Function component; by default the component displays the report when it is executed. The "display" takes either the form of a window where the report is shown (foreground execution) or of a saved entry in the configuration database that is shown in the Saved Reports or Saved Processes window (background execution). Every time the function is executed (that is, on every pass through the loop), a new window/saved report entry is created.

If the sole purpose of the report is to be e-mailed, as in this example, the display of the reports can be suppressed by leaving the "Display report" checkbox in the Function Inspector unchecked.



The Function component that encapsulates the Report Procedure returns the Report variable that carries the PDF-formatted report. This variable is then fed into the SendEMail System function.

The SendEMail function takes a number of parameters that are set to the procedure variables as follows:

<i>Parameter</i>	<i>Set to</i>
Mail server	Name of the mail server to which the e-mail will be sent. In our example it is set to a fictitious "our.mail.server".
To	E-mail address of the recipient. Set to the e-mail address returned from the Customers table.
From	E-mail address of the sender. In our example all e-mails are sent from the fictitious "manager@northwind.com".
Subject	Text to be shown in the e-mail subject field. In our example is set to "Your order history".
Message	Text to be sent in the body of the e-mail. In our example the message is formatted in the Calculation component.
Attachment	The report carried by the Report variable, and formatted as PDF.
Error	Error message if problems with sending the e-mail arise

The SendEMail function returns an INTEGER variable that is set to 0 if the e-mail was sent successfully. We do not check it in this procedure, so it is called "Dummy", but in a real-life situation we really should, in order to keep track of failed messages for subsequent research and re-run.

Save the Process Procedure as MailCustomerOrderInfo.

HTML formatting

Reports can also be e-mailed as HTML-based messages. If the REPORT variable is fed into the "Message" slot of the SendEMail function instead of the "Attachment", the report it carries will be formatted as HTML. In this case the "Attachment" can be left empty.