User Interface - Feature #12188

server-side HTML-to-PDF rendering callable from 4GL

11/16/2026 03:31 PM - Greg Shah

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**reviewer:**	
vendor_id:	GCD	**production:**	No
case_num:		**env_name:**	
version_reported:		**topics:**	
version_resolved:			

Description

History

#1 - 11/16/2026 03:31 PM - Greg Shah

Problem

GUI applications often print HTML documents (letters, mail merge output, formatted notes) by loading them into a Windows browser control and calling its print method, or by handing an HTML file to the Windows shell with the print verb. In a converted system the 4GL logic runs on the server and the user is at a browser, and the accepted way to print is to deliver a PDF to the browser with OPEN-MIME-RESOURCE. FWD currently has no facility to turn HTML into PDF on the server, so each project has to find its own library.

A batch process such as a mail merge also needs to combine many documents into a single PDF, so the user prints one file rather than receiving one browser tab per document.

Work

1. Choose the rendering library (for example openhtmltopdf over PDFBox, or Flying Saucer), considering HTML and CSS coverage, licence, font handling, and how it fits FWD's existing PDF and font infrastructure. Document the HTML and CSS subset supported.
2. Provide the 4GL-facing API. Proposed: a built-in FWD class (for example com.goldencode.p2j.util.HtmlToPdf exposed through Direct Java Access, or a 4GL-callable built-in class) with:
   - render an HTML file or a LONGCHAR to a PDF file;
   - a base URL or directory for resolving relative links to CSS and images;
   - page size, orientation and margins, defaulting to FWD's print options (as used by the GUI PDF printer);
   - append several documents, each starting on a new page, into one PDF;
   - an option to embed fonts, and font mapping consistent with FWD's PDF printer.
3. Resolve images and stylesheets from local files, the server's file system mappings, and http/https URLs (with a timeout and an option to disable remote fetches). Report missing resources in the log without failing the document.
4. Character encoding: honour the document's declared charset, and the session codepage for LONGCHAR input.
5. Errors: invalid HTML is rendered as well as the library allows; fatal errors raise a 4GL error that can be trapped with CATCH or NO-ERROR.
6. Document how to deliver the result with OPEN-MIME-RESOURCE, and an example of a batch that combines many documents.

Testcases

- A letter with a linked stylesheet, a local image, a remote image and a table renders with correct layout and fonts.
- A LONGCHAR with non-ASCII characters, in ISO8859-1 and UTF-8 sessions.
- 500 merged letters appended into one PDF, each on its own page; memory use stays bounded.
- A4 and Letter, portrait and landscape, custom margins.
- A missing image (logged, document still produced) and malformed HTML.