PDF Cover for Total CMS 3 - Version 2.1.0

PDF Cover creates a cover image from page 1 of a PDF in the browser with Mozilla PDF.js. It integrates automatically with configured TCMS admin fields and also provides a reusable JavaScript API for custom pages and ordinary HTML forms.

No executable as in version 1, `exec()` call, Poppler installation, server-side rasterizer, or CDN is required.

Schema configuration

Add `pdfCover` to the settings of the destination image field. `source` is the property name of the source file field.

```
{
  "file": {
    "type": "file",
    "field": "file",
    "settings": {
      "rules": {
        "accept": ["application/pdf"]
      }
    }
  },
  "cover": {
    "type": "image",
    "field": "image",
    "settings": {
      "pdfCover": {
        "source": "file",
        "maxWidth": 600,
        "quality": 0.85,
        "format": "webp"
      }
    }
  }
}
```

Options

Option	Default	Description
`source`	required	Property name of the TCMS `file` field containing the PDF.
`maxWidth`	`600`	Output width in pixels, limited to 100–4000. Smaller PDF pages are not enlarged.
`quality`	`0.85`	WebP/JPEG quality, limited to 0.1–1.
`format`	`webp`	`webp` , `jpeg` , or `png` . If WebP encoding fails, JPEG is used.

The existing `aiace-vox` configuration (`file` → `cover` , width 600) is compatible.

Workflow

1. Open a new or existing object in the TCMS admin.
2. Select a PDF in the configured source field.
3. PDF Cover renders page 1 locally and inserts the result into the image field.
4. Save the object. TCMS saves the object and uploads both queued files normally.

A status message appears below the destination image field while the cover is being generated. Selecting another PDF replaces a cover previously generated by this extension; a cover manually selected afterward remains under the user's control.

Custom pages and ordinary HTML forms

The custom page must be served from the same TCMS origin so it can import the extension assets. Connect a PDF file input to an image file input with `bindPdfCoverForm()`:

```
<label>
  PDF
  <input id="publication-pdf" type="file" accept="application/pdf">
</label>

<label>
  Cover
  <input id="publication-cover" type="file" accept="image/*">
</label>

<p id="cover-status" hidden aria-live="polite"></p>

<script type="module">
  import { bindPdfCoverForm } from
    '/api/ext/eurosquare/pdf-cover/assets/scripts/pdf-cover-form.mjs';

  const disconnect = bindPdfCoverForm({
    source: '#publication-pdf',
    target: '#publication-cover',
    status: '#cover-status',
    maxWidth: 600,
    quality: 0.85,
    format: 'webp',
    onCover(coverFile) {
      console.log('Generated cover:', coverFile.name);
    }
  });

  // Call disconnect() if this form is removed from a single-page application.
</script>
```

The adapter assigns the generated image `File` to the target input and dispatches a bubbling `change` event. The page's existing form submission or upload code can then handle it as a normal user-selected image.

Direct JavaScript API

For a custom uploader, import the core function directly:

```
import { createPdfCover } from
  '/api/ext/eurosquare/pdf-cover/assets/scripts/pdf-cover-core.mjs';

const coverFile = await createPdfCover(pdfFile, {
  maxWidth: 600,
  quality: 0.85,
  format: 'webp'
});

const body = new FormData();
body.append('cover', coverFile);
```

`createPdfCover()` accepts a PDF `File` and returns a Promise containing the generated image `File`. Importing the core module does not scan or modify the page.

Pages on another domain

The example above is intended for pages on the same origin as TCMS. For a different domain, host the extension assets on that domain or configure an explicit CORS policy. Upload authentication, CSRF protection, and TCMS permissions still apply to the page's upload request.

Limits

- Automatic schema-driven generation runs in the TCMS admin. Other browser pages can invoke the public API described above.
- Imports performed entirely through the API, CLI, cron, or another server-side process cannot run this browser JavaScript.
- Password-protected or damaged PDFs produce an error in the form and are not converted.
- The PDF contents stay in the browser, apart from the normal upload performed by TCMS when the object is saved.

Bundled library

This extension includes `pdfjs-dist` 6.5.438 from Mozilla PDF.js. Its Apache-2.0 license is included under `assets/vendor/pdfjs/LICENSE`.