

Extensibility

This page is for developers and partners who build on top of DEXPRO Core. It describes Core's extension surface: the interfaces you implement, the extensible enums you add values to, the integration events you subscribe to, and the public codeunit APIs you can call.

Interfaces

DXP ISource Document

Implemented per **DXP Document Class** enum value to describe how a source document of that class is turned into processed data. Methods:

- **CreateProcessedJsonFromSource** — build the processed JSON from the raw source.
- **CreateSource** — create the source representation.
- **InitFieldMapping** — initialize the field mapping for the class.
- **IsSourceDataPlausible** — validate that the source data is plausible before processing.

DXP IDocument Processing

Bound per **DXP Target Document Process** enum value to perform the actual processing. Method:

- **ProcessStandardDocument** — process the standard document for the target process.

Extensible enums

- **DXP Document Class** — the classification of an incoming document (default *Unassigned*).
- **DXP Target Document Process** — the target processing to run.
- **DXP Next Process Step** — the next step for a document class.
- **DXP Document Status** — the lifecycle status of a Core document.

Add your own values to these enums from a dependent app to introduce new document types, processes, steps, and statuses.

Integration events

Key events are published from **DXP Document Mgt.** and **DXP Document Transfer Mgt.**:

- OnAfter / OnBefore purchase **header** creation.
- OnAfter / OnBefore purchase **line** creation.
- Events around **dimensions** handling.
- Events around document **release**.
- **OnResolveDocumentViewerUrl** — partners supply a viewer URL for an attachment/document.
- **OnBeforeLookupPageFromPlausibilityEntry** — influence the lookup page opened from a plausibility entry.

Subscribe to the OnBefore events to adjust or skip default behavior, and to the OnAfter events to enrich the created records (for example to populate custom fields not covered by field mapping).

Public codeunit APIs

- **DXP Document Mgt.** — add and update Core documents (*AddDocument*, *UpdateDocument*), move a Core document (*MoveCoreDocument*), and run duplicate checks.
- **DXP Rule Engine** — evaluate rule sets from your app.
- **DXP Web Request Mgt.** — shared HTTP client for outgoing calls (honors the HTTP Timeout in Core Setup).
- **DXP OAuth2 Mgt.** and **DXP OpenID Mgt.** — obtain tokens for configured OAuth 2.0 / OpenID Connect connections.
- **DXP Master Data Mgt.** — work with vendor master data.
- **DXP Telemetry** — the shared telemetry helper for logging feature usage and errors.

Typical extension pattern

- Add a value to **DXP Document Class** for your new document type and implement **DXP ISource Document** for it.
- Add a value to **DXP Target Document Process** and implement **DXP IDocument Processing** to process it.
- Configure the **Next Process Step** for the class in Document Class Setup.
- Subscribe to the relevant OnBefore/OnAfter events to customize record creation.
- Call the public codeunit APIs (Document Mgt., Rule Engine, Web Request Mgt., OAuth2/OpenID Mgt., Master Data Mgt., Telemetry) as needed.

Related features

- Document Class Setup
- Field Mapping
- Rule Engine
- OAuth 2.0 and OpenID Setup

Revision #1

Created 2026-07-22 07:21:21 UTC by Bernd Feddersen

Updated 2026-07-22 07:21:21 UTC by Bernd Feddersen