

# BlockDescriptions

This page contains descriptions of the various blocks and what they do/provide.

Eventually they may be broken out into separate pages but let's keep them here for now.

Block dependencies are listed in blocks.properties as well, please update or inform the dev mailing list if you find new dependencies.

---

## apples

*Apples is a new flow implementation for Cocoon that allows you to write your deciding flow components (business logic) in pure OO Java.*

## asciiart

*Provides a generator for reading ASCII art files and rendering them into images.*

## authentication-fw

*Components for the authentication framework. Provides LoginAction, LogoutAction, ProtectedAction, ConfigurationGenerator. Include samples.*

Requires: session-fw block.

## axis

*Provides the AxisRPCReader, which accepts a SOAP request. Allows you to serve SOAP requests from your Cocoon application.*

## Batik

*Allow rendering [SVG](#) images to JPEG and PNG using [Batik](#) library. Include SVGSerializer, FragmentExtractorGenerator and FragmentExtractorTransformer. Includes samples. Required-By: fop block*

## BeanScriptingFramework

*\_ The Bean Scripting Framework (BSF) incorporate scripting into Cocoon. Using BSF, an application can use scripting, and become scriptable, against any BSF-supported language. When BSF supports additional languages, the application will automatically support the additional languages. Provides ScriptAction and ScriptGenerator. Include the required bsf libraries.\_*

## chaperon

*Provides TextGenerator and the required Chaperon libs for the creation of XML from a structured plaintext (Grammars). See also [Chaperon @ SF](#)*

## cron

*Provides a cron scheduler. Helps to run some task at defined intervals of time. Eg: every day at 4:00 a.m. run maintenance of the database*

Required-By: scratchpad block

## databases

*Provides various database-related actions and transformers, [ESQL](#) and other database-related logicsheets, as well as various utility classes. You probably want this if you desire any sort of database connectivity.*

## deli

*\_Provides deli and rdfilter libs, DeliTransformer, as well as the custom Deli components that Cocoon needs.*

For more information, see the [Deli](#) documentation.\_

## eventcache

*Event-based cache invalidation, where the events are external to Cocoon, such as a database update. See the samples for more info.*

## FOP

*FOP* reads an [XSL-FO](#) document and turns it into a PDF, PS or PCL format document using *FOPSerializer*. Include libraries and docs. Include FOP libraries, components and samples.

Requires: Batik block

## hsqldb

Add support for [HSQL Database Engine](#) a 100% Java Relational Database.

## html

Provides the html file generator which extract xhtml sax events from any html file. It is based on [jtidy](#).

## itext

You need [iText](#) if you don't want to use the FOP block to generate PDF.

## jfor

Allows you to serialize data as RTF using FO.

## jsp

Provides the JSPGenerator, JSPReader, utility components. Include samples.

## jxforms

Provides a form framework based on [xform](#). Include samples.

## linkrewriter

Provides the *LinkRewriterTransformer* and the *VariableRewriterTransformer*. These allow you to change links in your pages, for somethin as simple as adding a request parameter to every URL, or something more complicated as changing links to point elsewhere. See the samples and the Javadocs for the components for more information.

## linotype

*Linotype* is a robust weblogging tool written on top of Cocoon. Check out the samples for more info.

## lucene

Provides the components needed to add full text search to your site, including the [LuceneIndexTransformer](#). See also [the Lucene website](#)

## mail

Functionality for sending mail using the *SendmailAction*, a *Sendmail Logicsheet* for your XSPs, and an experimental *IMAPGenerator*. There is also a *WebMail* application included in the samples for this block.

## MIDI

The MIDI block provides components to generate XMidi (an XML representation of MIDI) from a MIDI file, and to serialize XMidi back to a MIDI file. There are also simple stylesheets for transposition and inversion.

## naming

Provides the *LDAPTransformer*, for LDAP access. "Naming" comes from the N in [JNDI](#).

## OJB Block

Provides components to use [Apache OJB](#).

## petstore

*The Cocoon flowscript version of the J2EE PetStore.*

## php

*Provides the [PHPGenerator](#). This block is considered **deprecated in the 2.1.x branch** of Cocoon and is **removed in 2.2-dev**. The servlet that is distributed with PHP never worked correctly.*

## poi

*Provides a [HSSF](#) serializer that produce MS Excel spreadsheet format output. The input format must be the one used by [Gnumeric](#) .*

## portal

### portal-fw

*Provides the components which compose the framework which allows you to build a portal with Cocoon. Deprecated.*

## precept

*Form processing. Check the README file included with this block for more information.*

## profiler

*Provides the ProfilerGenerator, which allows you to debug and profile your pipelines.*

## proxy

*Provides some generators which proxy requests to remote HTTP servers, including the HttpProxyGenerator and the WebServiceProxyGenerator.*

## python

*Enables to use the Python language in XSP pages*

## qdox

*Parse JavaDocs using [QDox](#).*

## samples

*Add the samples of the Cocoon distribution.*

## scratchpad

*new components or core functions under experimental development.*

Requires: cron block

## session-fw

*Provides the components to manage (create, modify, delete) sessions in xml with the Session Transformer.*

Required-By: authentication-fw block

## slide

*Jakarta Slide integration.*

## slop

*\*Simple Line Oriented Parser. Allows you to parse text files using the SlopGenerator.\_*

## stx

*Allows you to use [STX](#) for transforming your documents.*

## swf

*Provides the SWFGenerator and SWFSerializer for the creation and manipulation of Flash animations using Cocoon. Includes the required [Spark](#) library.*

## taglib

## velocity

*\_ This is a generator that uses the velocity template engine to feed velocity documents into the Cocoon pipeline. See [Velocity.\\_](#)*

## web3

*Enables Cocoon to communicate with SAP R/3(TM). Further information can be found in the documentation.*

## webdav

*Provides interconnectivity for [WebDav](#). Allows you to use WebDav sources and also provides the DASLTransformer. Check out the samples for more info.*

## Woody

*Provides a form framework. Include samples. Deprecated.*

## xmlldb

*Provides infrastructure for connecting to XML Databases using a number of methods, including the XMLDBGenerator, XMLDBCollectionGenerator, DbXMLAuthenticatorAction, and the [XMLDBTransformer](#).*

## xmlform

*Form processing. Deprecated.*

A picture of cocoon blocks with a lot of pointers for dependencies [blocks-218.ppt](#)