MooD International Support Portal Support Solution

**MDD - Explain briefly how it works.**

A model driven document (MDD) is a Microsoft Word® document (**.doc**) generated from a MooD repository. Each MDD is a type of MooD synchronizer called **Documents**. This defines what is extracted from the repository and how it is structured and formatted. For example, you could create an MDD that generates a document detailing all projects with a particular status.

Once you have a Documents synchronizer, you use it like any other synchronization you have in Business Architect. Synchronizers are listed under **Synchronize** in the Explorer Bar’s **Libraries** section, and you can schedule or execute them as required.

Each MDD is based on a Microsoft Word template. A document based on this template is created and fragments are inserted into it. Patterns are used within fragments. Put simply, fragments are the elements you want to include in the document, and patterns are the content you want to include for each element. The sections that follow elaborate this basic description.

# About the Microsoft Word template

All generated documents (**.doc** files) are based on a Microsoft Word template file (**.dot** or **.dotx** files). The template defines the basic properties of the document such as page size, margins, headers and footers etc. and any fixed text required, such as corporate logos, titles, or standard content. The template also defines the Microsoft Word styles that you will use in patterns.

The template can include bookmarks and you can insert content from the MooD repository directly into a bookmark location. Bookmarks can be included in the body of the template and in its headers and footers. One use of bookmarks is to have an entire document as the underlying template and only insert specific pieces of information from the MooD repository at bookmarked points. This is in contrast to generating all of a document’s content from the repository. You can, of course, have any combination of fixed and generated content.

When you generate a document, Business Architect creates a new document based on this template file. It will not alter the template file itself. This means you can use the same template file as the basis for many generated documents. You can create a Microsoft Word template file in a variety of ways, including saving an existing Word document as a template file. See your Microsoft Word documentation for more information on templates.

# About fragments

Fragments define structure and are inserted into the document.

Fragments reference elements in the MooD repository. Elements can be explicitly inserted or inserted as the result of a query. Fragments and their elements are inserted into the generated document.

Fragments provide the structure for the content that will be extracted from the MooD repository. You can have multiple fragments. Each fragment can reference as many individual or sets of elements as required.

MooD International Support Portal Support Solution

The order of fragments is the order they will be inserted into the document unless you use bookmarks in your template file. Fragments can be inserted at a bookmark location. If you use bookmarked fragments, any fragments that are not bookmarked are inserted at the end of the document.

# About patterns

Patterns define content and format. Patterns are used within fragments.

Each element is given a pattern. As elements are inserted into the generated document their patterns are applied. The pattern defines what content from the element will be added to the document, and how that content will be ordered and formatted (using styles from the underlying template). For example, a pattern can include the element’s name formatted as a heading, field values formatted as a list, and then an image of the element’s model. Business Architect gives you a variety of repository building blocks (***rules***) to construct your patterns, such as, Name, Field and Matrix. Patterns are reusable and ensure consistent content and presentation within the generated document.

# Exposing variables and parameters as pins

If you include content from your repository that requires variables, you can either set these at the rule level, or you can pass them upwards to the document level. At the document level they become exposed for pinning (and hence collected from the user).

***Readers Comments (if any) Images (if any)***