Ensemble Virtual Documents TechFAQ

An introduction to virtual documents

Ensemble virtual documents enable your productions to work with large and complex documents with little overhead.

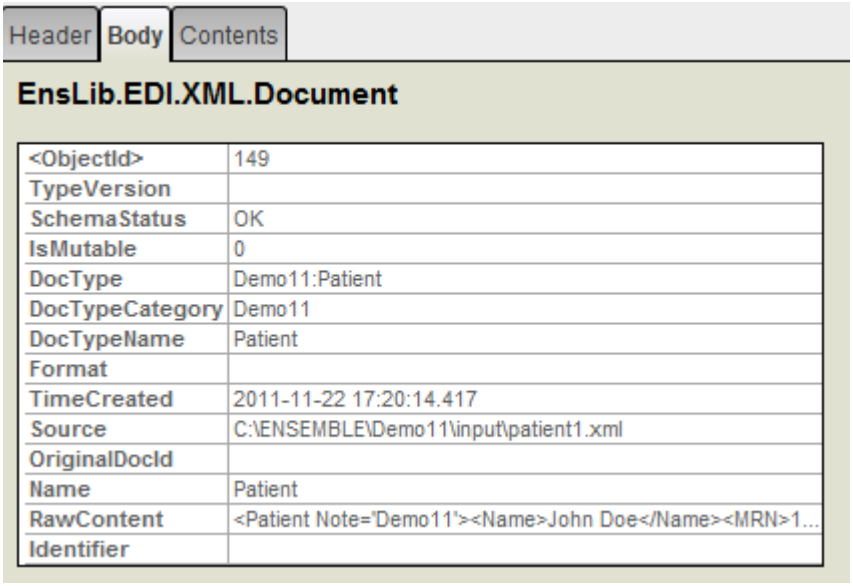
WHAT IS A VIRTUAL DOCUMENT?

A virtual document is a special kind of Ensemble message, for use with Electronic Data Interchange (EDI) messages and with XML documents.

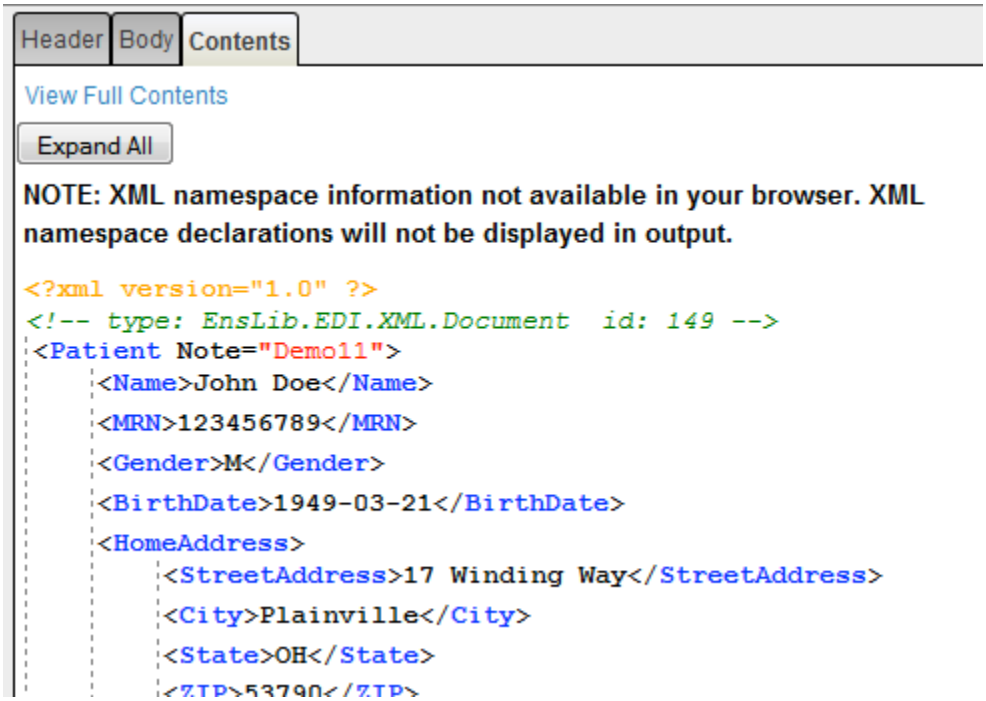
HOW ARE VIRTUAL DOCUMENTS DIFFERENT FROM STANDARD MESSAGES?

The body of any Ensemble message is represented by a persistent class that has one property for each value that the message is intended to carry. In addition to carrying these values, the properties enable users to easily search the messages that have passed through the production. Specifically, the message search option of the Management Portal automatically provides convenient drop-down lists of the properties.

For a virtual document, the message contents are stored within a global. The message body provides access to the message contents via a mechanism described later in this article. The message body is persistent and includes a small set of properties to carry information about the associated virtual document. The following shows an example:



The message body also includes a computed, transient property (RawContent) that the Management Portal uses to display the message contents in truncated form. This property is shown above and is also shown on the Contents tab of the message trace:



WHAT ADVANTAGES DO VIRTUAL DOCUMENTS PROVIDE?

Virtual documents are useful when Ensemble needs to examine only a few pieces of a large or complex message.

EDI message formats are designed to facilitate the transfer of large and complex electronic documents between different applications. The formats are correspondingly complex. A classic Ensemble message body would require a message class with hundreds of properties. Creating an instance of such a class can never be as rapid as creating an instance of a virtual document class, which has fewer properties.

Similarly, XML documents can be quite complex. Although an XML document can be easily represented as a set of XML-enabled classes, instantiating multiple classes takes more time than instantiating a single virtual document class. Furthermore, if you expect your production to receive XML documents of forms that are unknown in advance, virtual documents can carry the documents without your needing to specify the forms that your production can receive.

HOW DO I ACCESS VALUES IN A VIRTUAL DOCUMENT?

You start by defining virtual properties. Each virtual property has a name and is defined as a specific location (or *property path*) in a document. You define the virtual properties that you need, and these properties are indexed in the same way as properties of standard message bodies. You use the virtual properties within data transformations and business rules in the same way that you use properties of standard message bodies.

The details for property paths depend on the kind of virtual document. Each EDI format has a standard that specifies how each structure can be parsed. Correspondingly, for each EDI format, Ensemble provides a suitable syntax that reflects how that kind of document can be parsed.

Similarly, Ensemble provides syntax for accessing contents of XML documents in a general way. You can access any part of any XML document, no matter what XML schema it complies to.

In either case, to retrieve values from and set values in the virtual document, you use the usual GetValueAt and SetValueAt methods of the message body, and you provide a virtual property path as a method argument.

How Do I Indicate which Kind of Property Path I am using?

Obviously, with multiple possible syntax variations, it is necessary to indicate which syntax is in use in any given scenario. Ensemble uses the following system:

- Each general kind of virtual document is represented by a different subclass of Ens.VDoc.Interface. For example, EnsLib.HL7.Message can represent any HL7v2 message as a virtual document. In this class, the GetValueAt and SetValueAt methods recognize property paths that use the syntax that Ensemble provides for HL7v2 messages. Similarly, EnsLib.EDI.XML.Document can represent any XML document. The GetValueAt and SetValueAt methods of this class recognize property paths that use the syntax that Ensemble provides for XML documents.

- Ensemble also provides business service and business operation classes that are hardcoded to use specific message classes. For example, EnsLib.EDI.XML.Service.FileService sends messages that are instances of EnsLib.EDI.XML.Document.

- When you add a business service or operation, you choose a class. This choice indirectly specifies the message class for that business host to expect and to use.

  In addition, as a configuration option, you indicate the specific message type to expect (such as ADT_A01). This provides Ensemble with complete information about how to interpret any property paths used within this business host.

  For XML documents, you can specify the XML schema to use, but doing so is optional.

- For a business process, Ensemble provides a specialized routing engine to handle virtual documents. If you base a business process on EnsLib.MsgRouter.VDocRoutingEngine, you can use virtual property paths in any rule sets and data transformations used by this business process. In these rule sets and data transformations, you specify the message class to expect (such as EnsLib.HL7.Message or EnsLib.EDI.XML.Document), and you indicate the specific message type to expect. For XML documents, it is optional to specify the type.

How can users access values in virtual documents?

When searching for messages in the Management Portal, a knowledgeable user can directly type virtual property paths. However, it is probably more common to define search tables for the benefit of the users. Then, to search, users can choose from a list of searchable properties.

A search table is a class that defines searchable properties. Its definition includes the type of virtual document (HL7v2 versus XML, for example), and the message type to which it applies (optional for XML documents). In the search table class, each searchable property is defined as a user-friendly name and a virtual property path.

As a configuration option, you specify the search table class for a business service or business operation to use. Then, as that business host processes messages, Ensemble uses the search table to index values in the messages, using the names you defined in that class.