XML Documents

XML documents are plain text files that contain information which has a hierarchical or tree structure. One example is a database, which has many records, each of which contains many data fields. Another is a configuration file, which may have several sections, each of which may contain sub-sections, which may contain values. The structure is represented using tags, like this:<info>Some information</info>Tags may have attributes as well as information:<info item="4">Some more information</info>In this example, "item" is an attribute of "info" which is set to 4. "Some more information" is the value of "info" item 4.Here is an example of an XML database:<Root>

```
<Data ID="1">
  <Name>John Smith</Name>
  <Age>43</Age>
  <SSN>234567</SSN>
</Data>
<Data ID="2">
  <Name>Jim Black</Name>
  <Age>29</Age>
  <SSN>223311</SSN>
</Data>
```

</Root>In this example, and the one before it, the attribute is used as a kind of record number. But in fact, it's entirely up to you how you structure your data. Where the amount of information is small (for example, numeric data) it is largely a matter of personal preference whether you store it as an attribute to a node, or as the values of sub-nodes.VDS XMLThe XML format is defined by international standards. It's advantages are readability, portability and platform independence. XML is commonly used as a portable, platform-independent way to exchange data between different systems on the Internet. XML's disadvantages are that it is not the most space-efficient way to store information, nor does it support fast ways of accessing it. Although XML can be used for databases, this is only practical if the database is small.The VDS implementation of XML is a "lite" implementation that is less restrictive than standard XML. In VDS, tags are not case sensitive, and tag names may be almost anything that does not contain a space, angle brackets or ampersand. Therefore it is up to you to keep to the standards if you want to create XML that can be read by other software. If your documents will only be used by other VDS programs you can be a lot less restrictive.The VDS XMLDOC command and @XMLDOC function are used to perform operations on XML documents. There is also a LIST SELECT command that can be used to select information from an XML database into a VDS string list.VDS XML documents are held in memory as a tree structure. All operations on an XML document are performed in memory. The XML document must be saved to disk to preserve any changes that have been made.XML pathsLocations within an XML document tree structure are called nodes. Nodes may simply be places to store information, or they may be used to store more complex hierarchical information by creating subnodes. It is important to understand this concept, because many of the operations carried out by the VDS XML commands and functions are performed relative to the currently selected node. This may be the root node, or a sub-node that has become the current node as the result of a previously executed function or command.The path to a node in an XML document is specified in VDS using a syntax similar to XPath (a syntax used by many XML implementations.)Where a command or function has a <path> parameter, the following rules apply:·If a path is relative to the root of a document it begins with a forward slash.·If a path does not start with a forward slash, it is relative to the current node.·If the path parameter is null, the current node is used.Note that the current node may only be used after a command or function that sets the current node. Commands and functions that set the current node include: XMLDOC ADD, @XMLDOC(SEEK, &hellip;), @XMLDOC(FIRST, &hellip;), @XMLDOC(LAST, &hellip;), @XMLDOC(NEXT, &hellip;) and @XMLDOC(PREVIOUS, &hellip;).XML queriesA path may not uniquely locate a particular item of information in an XML document. To locate a particular record or group of records in an XML database, or a particular item of information, a simple query syntax may be used. It is best illustrated by example:/Data/Person[Name=John]Select the Person node that contains a Name node with the value "John"./Data/Person[SSN=12345]/NameSelect the Name subnode of the Person node that contains an SSN node with the value "12345"./Data/Person["@ID=3"]Select the Person node with an "ID" attribute of "3". The "@" indicates that ID is an attribute of person, not a subnode. Note that because VDS usually interprets an @ as the start of a function name, you need to put quotes round this query./Data/Person[3]Select the fourth Person node (the first would be Person[0]).The operators that may be used in a query are:=Equal==Exactly equal (case sensitive){}Contains{=Starts with<>Does not equal>=Greater or equal>Greater than<=Less or equal<Less thanUnless specified, comparisons are not case sensitive.VDS does not support compound operations (in other words, combining operations with "and" or "or".) However, concatenating two queries, like this:/Data/Person[Name{}Smith][Age>50]has the same result as an "and" combination.