Using InterSystems Documentation

Version 2009.1
30 June 2009

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

Using InterSystems Documentation
Caché Version 2009.1 30 June 2009

This book was assembled and formatted in Adobe Page Description Format (PDF) using tools and information from the following sources: Sun Microsystems, RenderX, Inc., Adobe Systems, and the World Wide Web Consortium at www.w3c.org. The primary document development tools were special-purpose XML-processing applications built by InterSystems using Caché and Java.

InterSystems CACHÉ® and InterSystems ENSEMBLE®

Caché WEBLINK, Distributed Cache Protocol, M/SQL, M/NET, and M/PACT are registered trademarks of InterSystems Corporation.

INTERSYSTEMS, InterSystems HEALTHSHARE™, InterSystems DEEPSEE™ and InterSystems TRAKCARE™

InterSystems Jalapeño Technology, Enterprise Cache Protocol, ECP, and InterSystems Zen are trademarks of InterSystems Corporation.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Customer Support
Tel: +1 617 621-0700
Fax: +1 617 374-9391
Email: support@InterSystems.com

Table of Contents

List of Figures

About This Book

If you are reading this page, you are already using the InterSystems product documentation. However, there are some features that you might not expect, or use every day, that are very helpful when you need them. The chapters in this book describe:

- Basic Features
- Using the Search Engine
- Printing the Documentation
- Extending the Documentation
- Legacy Documentation

For a detailed outline, see the Table of Contents.

1

Basic Features

InterSystems provides documentation in HTML format and in Adobe Page Description Format (PDF):

- The Online Documentation can be displayed any time the InterSystems server is running.
- Class Reference information is available from the online documentation home page.
- PDF files of the online documentation can be viewed or printed as you choose.

1.1 Online Documentation

Whenever the local server is online, it displays documentation in HTML format via a Web browser. The online documentation system gives access to information about core InterSystems technologies and specific InterSystems products and features, such as Caché, Ensemble, HealthShare, Zen, DeepSee, Zen, and TrakCare.

1.1.1 Starting Up

While the server is running, you can access the online documentation in any of the following ways:

- Find the InterSystems cube in the Windows system tray. Click on it to display the cube menu, then choose **Documentation** from the menu.
- From the Windows **Start** menu, choose **Programs**. Navigate to the InterSystems product name and server instance name, then choose the **Documentation** option.
- Enter the following URL into a browser page on the local InterSystems server, where 57772 is the Web server port number configured for the server:

  http://localhost:57772/csp/docbook/DocBook.UI.HomePageZen.cls

- While using the System Management Portal, click the **Help** link in the banner at the top of the page. A second window pops up. If this does not already contain the information you need, click the **[Documentation]** link in the navigation bar near the top left of this page.

If your preferred InterSystems instance is not running when you request **Documentation** using any of these techniques, the documentation system responds as if you are having difficulty running that instance. That is, an HTML page displays from which you can navigate to the InterSystems documents that are most likely to help when the system is down. To view these documents, click on a language name in the page (**English** or **Japanese**) and choose a document from the list. These documents are in PDF format so that they can be viewed even when the server is down. If you want to see the full set of documents in HTML format, restart the preferred instance and request **Documentation** again.

Studio offers two convenient short cuts for displaying online documentation:

- While viewing code, select any text string in the source code and press the **F1** key. The online documentation **Search Results** page opens to reveal the results of a search on this text string. If the highlighted text string was an ObjectScript language keyword, the documentation for that specific item displays.
- From the Studio **Help** menu, choose the **On-line Documentation** option. This opens the online documentation home page, from which you can navigate to a specific book or topic.

If you have tried all of these techniques and still cannot view the online documentation, contact the InterSystems Worldwide Response Center for assistance. Meanwhile, you can still access InterSystems documentation in PDF format as described in the chapter "Printed Documentation."

1.1.2 Navigation

Each page of the InterSystems documentation system displays a banner at the top of the page that offers the following choices from left to right:

Option	Action
Home	Takes you to the main Documentation page.
Management Portal	Takes you to the System Management Portal home page.
Index	Takes you to the master index for the online documentation.
«	Displays the previous chapter or reference page in the book that you are currently viewing. Appears dimmed if there is no previous chapter or reference page. The **«** link is not the same as the Back button in your browser; it reflects book sequence rather than your browsing history.

Option	Action
»	Displays the next chapter or reference page in the book that you are currently viewing. Appears dimmed if there is no next chapter or reference page. The **»** link is not the same as the Forward button in your browser; it reflects book sequence rather than your browsing history.

Below the banner, a navigation bar contains the following items from left to right:

Option	Action
Navigation trail for the documentation system	To return to the main Documentation page, click **[Documentation]** in this trail.
Go to	The **Go to** box is visible only when you are currently viewing a book. It does not appear on the main Documentation page. The **Go to** box contains a list of section titles within the current page. Selecting one brings you directly to that section.
Search	Typing a word into the **Search** field and pressing the **Enter** key takes you to the **Search Results** page for that word. Alternatively, you can click on the **Search** label to go to the advanced search page. For details, see the chapter "Using the Search Engine."

Below the navigation bar, a menu offers the following choices from left to right. These choices are not visible when you are viewing a book. You can only use them from the main page:

Option	Action
Getting Started	Displays the official Getting Started page for the product.
Master Index	Takes you to the master index for the online documentation.
Feature Map	Displays a visual, schematic guide to the online documentation.
Class Reference	Opens the Class Reference documentation system.
Search Page	Invokes the Search Facility for online documentation.

1.1.3 Security

Security features can affect your ability to view the online documentation.

For example, depending on the security settings for your InterSystems server, you might need to perform special configuration to allow *any* user to gain access to the online documentation. For background information see the "Tightening Security for a Caché Instance" chapter in the *Caché Security*

Administration Guide. For a detailed procedure, see the section "Making the Documentation or Samples Available" in that chapter.

Also, on Windows, when trying to access the documentation via the InterSystems cube, the user ID assigned for the attempt is UnknownUser. This can cause problems in using the online documentation as follows:

- When the InterSystems server instance is installed with a security level of Normal or Locked Down, the UnknownUser only has the **%DB_DocBook:R** privilege. This is insufficient to read the Caché class reference documentation. The **%Development:Use** privilege is required.
- In order to run program examples in the online documentation, the user must have **%DB_SAMPLES:W** permission. If the UnknownUser lacks this permission, the button labeled **Run It** will not appear in any of the executable program examples.

The solution for both of these problems is to define one or more roles with the necessary permissions and assign these roles to the UnknownUser. Alternatively, you can edit the application definition for /csp/docbook to add these roles whenever it is run. See the "Roles" chapter in the *Caché Security Administration Guide*.

1.2 Class Reference Documentation

The online documentation home page displays a link called *Class Reference*. If you click this link, it displays the Documatic documentation system. This system contains the comments provided by InterSystems developers for the object-oriented classes in the InterSystems product code. Some elements of this code are private and cannot be seen in Documatic, but every item that can be made visible to customers is included here.

To view the *Class Reference* documentation, do one of the following:

- Start the online documentation system. On the main page, click *Class Reference*.
- Enter the following URL into a browser page on the local InterSystems server, where 57772 is the Web server port number configured for the server:

  http://localhost:57772/csp/documatic/%25CSP.Documatic.cls
- In Studio, From the **View** menu, choose the **Show Class Documentation** option.
- When viewing code in Studio, select the name of a data type and press the **F1** key. The *Class Reference* opens at the description of the class that defines this data type.

When the *Class Reference* main page displays, you can use the names in the left panel to navigate to the namespace, package, and class for which you want to view documentation. The right panel displays

the documentation. The *Class Reference* has its own banner which offers two choices from left to right:

Option	Action
DocBook	Takes you to the main Documentation page.
Search	Takes you to the Documatic search page.

When you add your own classes to your installation, comments you add to these classes using specific documentation conventions will appear in the *Class Reference*. For instructions, see "Extending the Class Reference."

The *Class Reference* has no PDF equivalent. It does have a search facility, which is entirely separate from the online documentation search described in the chapter "Using the Search Engine."

1.3 Printed Documentation

The full set of online documentation is available in PDF format.

InterSystems strongly recommends that if you want printed documentation, you *do not* print HTML pages from the browser, but instead take advantage of the pre-formatted PDF files, whose content is identical to the HTML. You can print these files yourself or send them to a print shop to be professionally printed and bound as books. For details, see the chapter "Printing the Documentation."

The full set of PDFs is available on your distribution DVD. Insert the DVD into your machine, and open it as a normal device. If the installer attempts to run, indicate that you do not wish to install anything. At the top level of the DVD is a directory, docs\full, whose subdirectories contain all the PDFs.

Many InterSystems products share core underlying technologies, so you might need to examine more than one subdirectory to find the documents you need. For example, the Ensemble product offers docs\full\ensemble for Ensemble information and docs\full\cache for technologies that Ensemble and Caché share.

A limited set of PDF files for an InterSystems product is also available under the directory where you installed the product, usually C:\InterSystems\Ensemble\Docs or similar. This is the documentation subset that can be referenced when the server is not running.

You can also download PDFs from the InterSystems Web site as follows:

1. Go to the URL www.intersystems.com/support
2. In the list of links at left, choose one of the two products — Caché and Ensemble — that allow you to download documentation as PDFs. The link labels are:
   - **Caché Support**

- **Ensemble Support**

3. In each case, when you click the link a page displays detailed information about support. Scroll down the page to find a link that you can click to view or download the documentation.

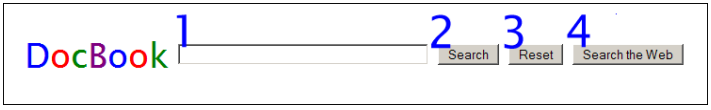
2

Using the Search Engine

To perform a simple search in the InterSystems documentation, enter the subject of the search in the field near the top right of the page next to the **Search:** label and then press **Enter**.

For advanced search facilities, click on the **Search:** label itself. The **Keyword Search** page displays the following fields:

Fields in the Keyword Search



1. The contents field — Enter search terms here.
2. The **Search** button — Click to search.
3. The **Reset** button — Click to empty the contents field.
4. The **Search the Web** button — Click to use the terms specified in the contents field for a Google search of the Web.

Note: You can read and navigate the online documentation if you have any level of database access, but in order to use the online documentation **Search:** feature, the account into which you are logged in must have Write privileges in the DOCBOOK database. Most accounts do have this access, but if you run into trouble with searches, you should adjust your privilege level, use a different account that does have the privilege, or contact the InterSystems Worldwide Response Center (WRC) for assistance.

2.1 Basic Search Rules

The following are the basic rules for searches:

- A search can include one or more terms.
- The search engine only returns paragraphs that include all terms. Such paragraphs can be titles, code fragments, and so on. (There is no "OR" search — that is, a search that includes either of two or more strings.)
- The order of search results depends on where the search terms appear. For example, a book title appears before a chapter title which appears before a paragraph of running text.
- The search is not case-sensitive. For example, searches for `SET`, `Set`, and `set` yield the same results.
- Some search results begin with one or more highlighted links, which have been flagged as notable results for the search term. Exact search (described in the next section) disables this feature.

2.2 Exact and Free-Text Searches

By default, the search engine uses the Caché free-text search (sometimes called a "stemmed search"). This kind of search is based on an extensive list of related forms of words. For any word appearing in the list, the search engine takes the stem of each word provided and finds all matches associated with that stem. For example, `save`, `saving`, and `saved` all yield the same results. The order of the results is the same for all variants of a particular stem (for example, regardless of whether a search is for `save` or `saving`).

The search engine supports stemmed search for most words. Words with forms that have definitions relevant to InterSystems products may not support stemmed search. For example, `include` and `including` return different sets of results, since "include" has a meaning in .

This is true for each term in the search, so that a search on `name convention limit` returns the section "Subscript Naming Conventions and Limits" because of stemming. For details on free-text search, see the section "Using Free-text Search" in the chapter "Querying the Database" of *Using Caché SQL*.

To search for an exact string, specify that string within quotation marks. This is known as an **exact search**. Exact searches are also case-insensitive. Therefore, `#SHOW` and `"#SHOW"` yield different results; however `"#SHOW"` and `"#show"` yield the same results. There is no search for strings that include quotation marks ("") or inch marks (").

Searches can include both exact and stemmed terms, such as a search for `saving files`, which returns results that stem both `saving` and `files`, versus a search for `"saving" files`, which returns results that include the exact string `saving` and any stemmed version of `files`.

2.3 Symbol Searches

The search treats all punctuation characters as white space (with the exceptions of "$" and "%"). For example, searches for `Sample.Person`, `Sample Person`, and `Person Sample` all yield the same results. This is also why searches for `#INCLUDE` and `include` yield the same results.

The search engine indexes words that include é as if they contained `e`; it also indexes words that include ñ as if they contained `n`. Hence, to search for `Caché`, simply enter `Cache` for the search; to search for the word "cache," use `"cache"`. Similarly, you can use `Jalapeno` instead of `Jalapeño`.

2.4 Interaction of Search Rules

The various search rules, taken together, offer a powerful means of locating information in InterSystems documentation. They can also sometimes cause unexpected search results. For example, the following are results for searches in Caché 2009.1 on "Mr." with various forms of punctuation and quoting:

Search String	Number of Results
`Mr.`	8
`"Mr."`	3
`Mr`	8
`"Mr"`	8

In this case, there is no stemming, so `Mr` and `"Mr"` return identical results. For `Mr.`, the period is converted to white space, which is then not included in the search, making it the same as searching for `Mr` (without a period at the end). For `"Mr."`, however, the search is for exact case of the characters M, R, and a period, in that order, and there are only three of these in the Caché documentation set. The difference of five hits arises from references the "MR" conversion code that appears in the MV Basic documentation.

3

Printing the Documentation

You can print any InterSystems PDF file while you are viewing it in Adobe Reader. However, to produce a useful paper document, with page numbers and headings arranged for easy skimming and information retrieval, InterSystems recommends the following procedure.

3.1 Print Options

To print and bind the PDF version of an InterSystems document, follow these steps:

1. Select a PDF file and copy it from the installation media to a local drive.
2. Send the PDF file to your administrative support team, or use a professional printing office. The cost for printing and binding as described in these instructions is usually quite reasonable, so InterSystems recommends using a professional print vendor. Most such vendors have a Web interface that allows you to upload your PDF file and specify where to deliver the paper document.
3. The person who prints and binds your document must choose among several options at each manufacturing step. The following table provides recommendations. The language used in this table is general and might vary slightly depending on the vendor.

Many print vendors are willing to bind a document that you have already printed yourself. You can output the printed pages using your own printer, then bring the pages to a print vendor for the cutting and binding. Doing this can save the apparent cost of the book, although you should check with your facility manager to see if the savings is real. The quality of the result in this case will depend on the quality of the printing machines at your office.

Choice	Recommendation
Page count	While viewing the PDF file in Adobe Reader or Adobe Acrobat, look for the total page count in the display.
Page range	Include all pages in the print job.
Double or single	Choose double-sided (duplex) printing if possible.
Ink color	You can print the books in color, but they print equally well in black-and-white: • When printing black-and-white, allow the normal shading options for gray tones. DO NOT request any specialized print option that outputs gray tones in all black ink. • If you want to print the book in color, consult your print vendor for the correct color options to choose.
Quality	Select the highest quality available (not "draft").
Resolution	Choose 600 dots per inch (dpi) or more.
Paper size	Book pages are designed to be 7 inches wide and 9 inches high, as seen on screen. However, you can easily print these pages onto A4 or letter-size paper as follows: • When printing the books in your office, while viewing the book in Adobe Acrobat, choose **File**, then **Print**. The Print dialog displays. Enable (check) **Auto-rotate and center pages**. Disable (uncheck) **Expand small pages to paper size**. Then you can print the book. • When printing with a vendor, find the space marked Special Instructions or Additional Comments on their print request form. Provide a note: "Image size is smaller than paper size. DO NOT expand image to fit page. DO auto-rotate and center pages."
Paper type	For black-and-white, choose a standard white paper. Some print vendors require high-quality laser paper for color; this affects cost, so check before you run the job.
Binding	Choose any type of binding you prefer: 3-ring, color coil, plastic comb, or other. Stapling is not suitable for these documents.
Covers	Many vendors offer a choice of cover colors and styles.

Choice	Recommendation
Cutting	You can ask the print vendor to cut the printed pages to 7 by 9 inches, as seen on screen in Adobe Reader or Adobe Acrobat. Cutting is not necessary, but it offers an inexpensive way to creates a more professional-looking result. If you want the pages cut, you *must*: • Print the book double-sided, *and* • Use the “Paper Size” instructions in this table, *and* • Specify how to cut pages, as follows: To achieve the intended 7 by 9 inch size, instruct the printer to cut pages by equal amounts top and bottom, and by equal amounts left and right. That is, for letter size paper, cut 1 inch from top, 1 inch from bottom, 0.75 inches from left and 0.75 inches from right.

3.2 Sample Print Request

The following is a typical print request for an InterSystems document.

Choice	Sample Entry
File name	GCOS.pdf
Page count	716
Double or single	Double
Ink color	Black-and-white
Covers	None
Paper size	Letter
Paper type	Standard
Binding	3–ring, or 3–hole punch
Special instructions	Image size is smaller than paper size. DO NOT expand image to fit page. DO auto-rotate and center pages.

4

Extending the Documentation

This chapter describes how to extend the online documentation system. There are two approaches:

- You can supplement the documentation with information about the classes you create, by placing InterSystems Documatic comments in your class code. These comments automatically appear in the online documentation system under the topic Class Reference.
- You can extend the formal Online Documentation by writing a new technical article.

4.1 Extending the Class Reference

You have the opportunity to add InterSystems Documatic comments each time you create a new item (such as a class, method, property, or query) using a Studio wizard. If you enter a block of text in the **Description** field of the wizard dialog, when you save your changes this text appears in the class in the lines immediately preceding the `Class`, `Method`, `Property`, or `Query` label. Studio inserts three slashes (`///`) at the far left of each line to indicate that it is a Documatic comment.

Once you compile the class, you can view its generated class documentation the next time you open the *Class Reference* documentation. If you add no Documatic comments, items that you add to a class or package appear appropriately in the lists of class or package contents, but without any explanatory text.

You can extend any existing Documatic comments from within Studio, either by editing the **Description** field for a class in the Studio Inspector window, or by adding specially formatted lines to the class code. The syntax rules for Documatic comments are strict:

- All Documatic comments that describe a class or class member must appear in a consecutive block immediately before the declaration of the item that they describe.
- Each line in the block of comments must start with three slashes: `///`

- The three slashes must begin at the first (left-most) position in the line.
- No blank lines are allowed within Documatic comments.
- No blank lines are allowed between the last line of the Documatic comments and the declaration for the item that they describe.

If you add Documatic comments using the **Description** field with a Studio wizard or in the Studio Inspector window, Studio handles these details for you. If you add Documatic comments directly into the code, Studio alerts you to some Documatic syntax errors: for example, if you insert a blank line between the comments and the declaration, or if you use an insufficient number of slashes at the beginning of a line within a Documatic text block. However, Studio does not alert you to any other types of bad syntax within Documatic comments.

Documatic comments allow plain text, plus any standard HTML element and a small number of specialized Documatic elements, as shown in the following code sample:

```
/// <p>Transforms <i>Star</i> order messages for <i>ChartScript</i>. <br/>
/// Developed Nov 2004 by <b>MT Engineering Team</b>. <br/>
/// See also <class>StarADTtoChartScript</class> and
/// <class>StarMRGtoChartScript</class> </p>
/// <p>Only Orders for these Departments pass: </p>
/// <ul><li>CP</li><li>NS</li><li>URO</li><li>NIV</li></ul>
/// <p>As long as they are one of the following:</p>
/// <ol>
/// <li>New Child Order</li>
/// <li>Child Order Status Change</li>
/// <li>Order Cancellation</li>
/// </ol>
/// <p>Data Transformation sets "T" in MSH 11 for Test environment.</p>
Class MT.dt.StarORMtoChartScript
      Extends Ens.DataTransformDTL [ ProcedureBlock ]

{
  // The data transformation class code goes here.
}
```

The previous example formats the *Class Reference* entry for the class as follows:

class **MT.dt.StarORMtoChartScript** extends Ens.DataTransformDTL

Transforms *Star* order messages for *ChartScript*.
Developed Nov 2004 by **MT Engineering Team**.
See also StarADTtoChartScript and StarMRGtoChartScript

Only Orders for these Departments pass:

- CP
- NS
- URO
- NIV

As long as they are one of the following:

1. New Child Order
2. Child Order Status Change
3. Order Cancellation

Data Transformation sets "T" in MSH 11 for Test environment.

With regard to the allowed HTML elements, adhere to as strict an HTML standard as you can, for example XHTML. This ensures that your comments can be interpreted by any browser. For a

description of specialized Documatic elements, such as <class>, <method>, <property>, <query>, <parameter>, <example>, and <link>, see the *Class Reference* for %CSP.Documatic.

4.2 Extending the Online Documentation

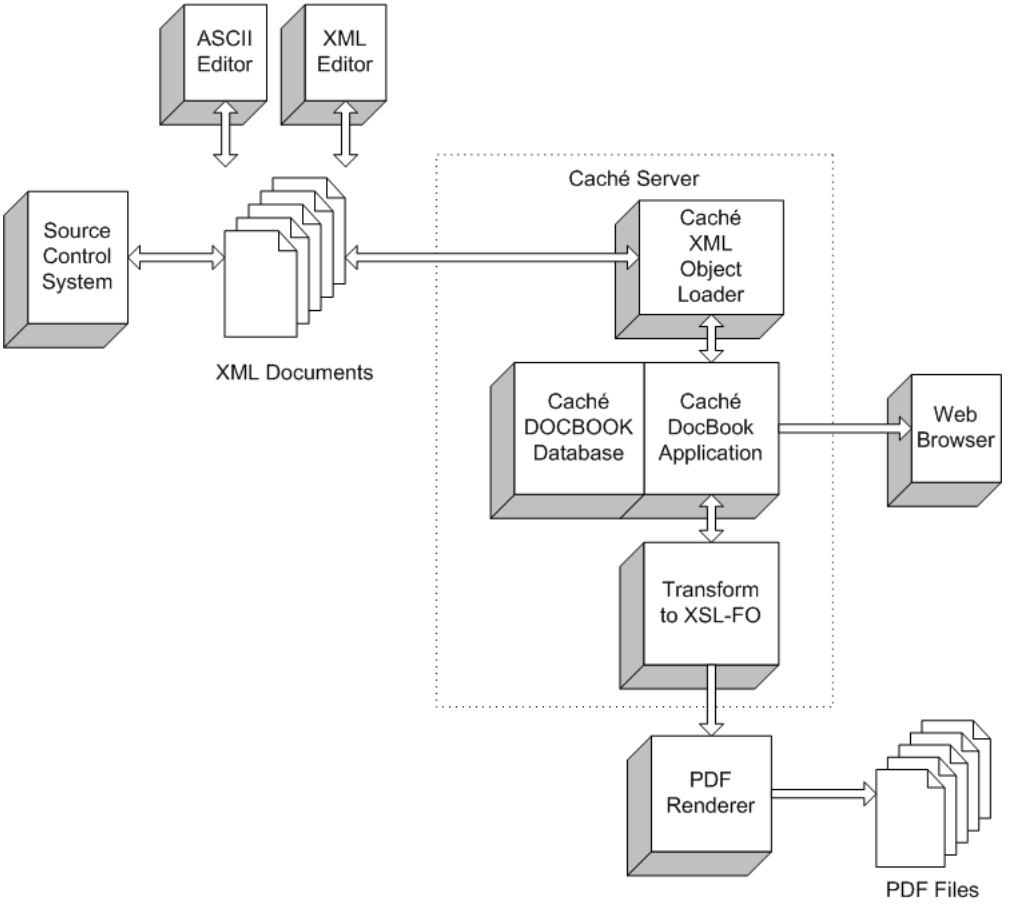
You can supplement the InterSystems online documentation system by adding technical articles to it. After you add them, readers can find these articles in any of the following ways:

- Start the main Documentation page. From the array of topics, choose **Articles**.
- Enter the following URL into a browser page on the local InterSystems server, where 57772 is the Web server port number configured for the server:

  http://localhost:57772/csp/docbook/DocBook.UI.Page.cls?KEY=KNBASE

4.2.1 The DocBook Application

The DocBook application consists of a database (DOCBOOK), installed as part of the InterSystems product installation. The DOCBOOK database contains the executable code for the application as well as the product documentation stored as a set of persistent objects. The following diagram summarizes how DocBook application formats and displays this documentation.



Note the XML documents in this diagram. Members of the InterSystems Documentation department edit XML documents that conform to the DocBook v4.0 standard. These files serve as the input for the Caché XML DocBook Loader, which transforms the files into HTML that is visible using a Web browser. To add a technical article that can be referenced by the InterSystems online documentation system, you must simulate the InterSystems Documentation department work process by creating your own XML documents that conform to the DocBook v4.0 standard.

4.2.2 The DocBook Standard

DocBook is an OASIS standard that defines an XML-based markup language for technical documentation. The installation of any InterSystems server includes a copy of the DocBook v4.0 XML DTD. For details about DocBook v4.0 refer to:

- *DocBook — The Definitive Guide* by Norman Walsh and Leonard Muellner, O'Reilly & Associates, 1999

- OASIS (Organization for the Advancement of Structured Information Standards) Web site: www.oasis-open.org
- DocBook web site: http://www.docbook.org/

4.2.3 Adding an Article to the Technical Articles

With a text editor, some knowledge, and something to say, you can easily create additional technical articles for inclusion within the InterSystems online documentation system. A specialized XML editor is helpful, but not necessary. Here is an XML document that defines an extremely simple article:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE article PUBLIC "-//Arbortext//DTD DocBook XML V4.0//EN"
"c:\cachesys\csp\docbook\doctypes\docbook\docbookx.dtd">
<article id="UMFA_MyFirstArticle" arch="intersystems">
  <title>My First Article</title>
  <subtitle>This document contains my first article.</subtitle>
  <para>This is my first article.</para>
</article>
```

This document contains the following:

1. A required XML header.
2. A required DTD specification indicating that this document is an article and identifying the location of the DTD file. Use the exact DOCTYPE statement shown in the example. The corresponding file need not exist on your system.
3. An <article> element delimiting the contents of the article. Note the attributes provided in the example:
   - The *id* value for this article must be unique across the entire database. For this purpose, InterSystems recommends that users begin the <article> *id* with the letter "U" which InterSystems documentation does not use.
   - The *arch* value for all customer articles must be set exactly as shown:

     ```
     arch="intersystems"
     ```
4. A <title> and <subtitle> describing the article.
5. A <para> (paragraph) containing the content of the article. A real article would probably have more than one paragraph.
6. When you save this document, save it to a file with the same name as the article's *id*. In this case: UMFA_MyFirstArticle.xml

To load a new document into the database, do the following:

1. Create a new DocBook article, such as the one described in the previous section.

2. Start the Terminal.
3. Switch to the DOCBOOK namespace where the DocBook database application is located:

```
ZN "DOCBOOK"
```

4. Invoke the **Load** method of the DocBook.Utils class using the full path and file name of your XML document. For example:

```
Do ##class(DocBook.Utils).Load("C:\myDir\UMFA_MyFirstArticle.xml")
```

5. Start the main Documentation page. From the array of topics, choose either **Articles** or **All Books**. Your article is now included in the list.

4.2.3.1 UTF-8 Encoding Preferred

The DocBook application checks whether or not the source XML documents are encoded as UTF-8. That is, it looks at the XML header for the *encoding* attribute:

```
<?xml version="1.0" encoding="utf-8"?>
```

If the UTF-8 declaration is not present in the XML header, the application raises an assertion violation. This warning is not fatal; assuming there are no other, fatal errors, the document contents are properly loaded.

You can easily transform document sources that are not already in UTF-8 to that format using XSL.

4.2.3.2 Content Size Limitation

The text contents of any single DocBook element cannot consist of more than 30,000 characters.

If you use XML entities such as `&` then each character in the entity reference counts toward the total. That is, `&` contributes 5 characters, even though it is only visible in the display as one & character.

If you use inline elements for styling, the characters required for the inline element count as well. So, for example, the following <para> content consumes 46 characters:

```
<para>This is <emphasis>really</emphasis> important.</para>
```

Whereas in the displayed output it appears to consume only 25 characters, as in:

This is *really* important.

4.2.4 Using DocBook Elements

If you would like to create an article that is more useful than the simple example in the previous section, you can use more DocBook elements within the body of the article. The following topics describe some of the more commonly used elements:

- Sections
- Lists
- Program Listings
- Tables
- Inline Elements

For a list of all the DocBook elements that you can use when adding an article to the InterSystems online documentation, see "All Supported DocBook Elements" at the end of this section.

4.2.4.1 Sections

You can include sections and subsections using the <sect1>, <sect2>, <sect3>, and <sect4> elements. These must be nested correctly. Indentation is optional, but helps to show the nesting relationships between elements:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE article PUBLIC "-//Arbortext//DTD DocBook XML V4.0//EN"
"c:\cachesys\csp\docbook\doctypes\docbook\docbookx.dtd">
<article id="UMFA_MyFirstArticle" arch="intersystems">
  <title>My First Article</title>
  <subtitle>This document contains my first article.</subtitle>
  <para>This is my first article.</para>
  <sect1>
    <title>This is Section 1</title>
    <sect2>
      <title>This is SubSection 1-A</title>
      <para>Text</para>
    </sect2>
  </sect1>
  <sect1>
    <title>This is Section 2</title>
    <sect2>
      <title>This is SubSection 2-A</title>
      <para>Text</para>
    </sect2>
  </sect1>
</article>
```

You can simply avoid assigning *id* values to any section element other than the top level <article> element. However, if you are an experienced DocBook user and wish to use a more advanced element such as <link> to provide internal cross-references to sections within your article, you can do so by establishing an *id* value for the section and using this as the *linkend* value in the <link> element.

Keep in mind that the DocBook application requires all *id* values within the same <article> to stem from the <article> *id* value. This stem must be identical to the XML source file name. To pull all of these requirements together, consider the following sample article.

Because of its <article> *id* value, this example must reside in the file UMFA.xml. Note the <article> *id* value, the <sect2> *id* value, and the <link> *linkend* value:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE article PUBLIC "-//Arbortext//DTD DocBook XML V4.0//EN"
"c:\cachesys\csp\docbook\doctypes\docbook\docbookx.dtd">
<article id="UMFA" arch="intersystems">
  <title>My First Article</title>
  <subtitle>This document contains my first article.</subtitle>
  <para>This is my first article.</para>
  <sect1>
    <title>This is Section 1</title>
    <sect2 id="UMFA_Critical_Info">
      <title>This is SubSection 1-A</title>
      <para>Text</para>
    </sect2>
  </sect1>
  <sect1>
    <title>This is Section 2</title>
    <sect2>
      <title>This is SubSection 2-A</title>
      <para>Text</para>
      <para>
        Also see <link linkend="UMFA_Critical_Info">SubSection 1-A</link>
      </para>
    </sect2>
  </sect1>
</article>
```

4.2.4.2 Lists

You can create ordered lists:

```
<orderedlist>
  <listitem><para>Item</para></listitem>
  <listitem><para>Another Item</para></listitem>
</orderedlist>
```

As well as unordered (itemized) lists:

```
<itemizedlist>
  <listitem><para>Item</para></listitem>
  <listitem><para>Another Item</para></listitem>
</itemizedlist>
```

Note that you must place a <para> (or other block) element within the <listitem> element. You can include multiple blocks within a <listitem> if you like.

4.2.4.3 Program Listings

You can use a program listing block if you want to display sample code:

```
<programlisting> Write "Hello"</programlisting>
```

The DocBook application renders this as a code listing.

You can syntax check and colorize your example by setting the *lang* attribute:

```
<programlisting lang="COS"> Write "Hello"</programlisting>
```

The available *lang* attribute values for <programlisting> include:

lang Value	Language
BAS	Caché Basic
CLS	Caché Class Definition Syntax
CLS!MEMBER	A member (property, method, etc.) of a Caché Class Definition
COS	Caché ObjectScript. ObjectScript examples need at least one space character at the beginning of each line to be syntactically correct.
CSP	CSP text
JAVA	A complete Java class definition.
JAVA!SCRIPT	JavaScript. No syntax checking or colorization is done.
NONE	This indicates that no syntax checking should be done on the example. Either the material has a deliberate error in it to illustrate a point, or it is incomplete as a program unit.
SQL	An SQL statement
XML	XML
XML!FRAGMENT	A well-formed, but not necessarily complete, piece of XML

The maximum length for any single line inside any <programlisting> block is 81 characters. This count includes the characters that comprise the <programlisting> and </programlisting> elements. To reduce the character count in the first or last lines of your example, you can place these elements on lines by themselves, as in the following example:

```
<programlisting lang="COS">
 Write "Hello! This maximum length includes one leading space for ObjectScript."
</programlisting>
```

4.2.4.4 Tables

The DocBook application supports simple tables defined using either <table> or <informaltable>. The only difference is that <informaltable> has no title.

```
<informaltable frame="all">
  <tgroup cols="2">
    <colspec colnum="1" colname="Param" />
    <colspec colnum="2" colname="Descrip" />
    <thead>
```

```
      <row>
        <entry colname="Param">Parameter</entry>
        <entry colname="Descrip">Description</entry>
      </row>
    </thead>
    <tbody>
      <row>
        <entry>X</entry>
        <entry>The X location.</entry>
      </row>
      <row>
        <entry>Y</entry>
        <entry>The Y location.</entry>
      </row>
    </tbody>
  </tgroup>
</informaltable>
```

4.2.4.5 Inline Elements

Inside a paragraph or table entry, you can highlight words or phrases using a variety of inline elements. Those most common of these include:

Element	Description	Example
<classname>	The name of a class	The %CSP.Page class
<command>	The name of a program or other operating system command	The UNIX **rm** command
<emphasis>	Italicized text	This is *really* important
<filename>	The name of a file	The Main.html file
<function>	The name of a function	Use the **$Data** function
<guibutton>	A button within a GUI	Press the **OK** button
<guilabel>	A label within a GUI	Enter this in the **Name** field
<guimenu>	A menu within a GUI	Available within the **File** menu
<guimenuitem>	An item within a menu	Within this menu, use the **Save** command
<methodname>	A method name	The **Print** method
<property>	A property name	The Name property
<quote>	A quotation	This documentation is “easy” to understand
<superscript>	A superscript	a^b

Element	Description	Example
<systemitem>	A security-related item such as the name of a privilege	You need **%DB_DOCBOOK:Read** to access DocBook
<varname>	A variable or argument name	*count* specifies the number of iterations
<userinput>	User input	Enter **Hang 100** from the command line

4.2.4.6 All Supported DocBook Elements

The following table lists all the DocBook elements that you could possibly use when adding an article to the InterSystems online documentation. For ease of reading, this table omits the surrounding angle brackets for each element: <abstract>, <anchor>, etc.

The following list is more than ample for writing technical articles. For the limited set of truly useful elements, return to the overview at the beginning of "Using DocBook Elements."

Index	Supported Elements
A	abstract, anchor, answer, appendix, application, article
B	book, bookinfo
C	caution, chapter, citation, citetitle, classname, colspec, command
D	
E	emphasis, entry, envar, errorcode, errorname, example
F	figure, filename, firstterm, formalpara, function
G	glossary, glossdef, glossentry, glosslist, glosssee, glossterm, graphic, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu
H	
I	important, indexterm, informaltable, inlinegraphic, itemizedlist
J	
K	keycap, keycode
L	link, listitem, literal, literallayout
M	methodname
N	note
O	orderedlist

Index	Supported Elements
P	para, part, partintro, preface, primary, productname, programlisting, prompt, property
Q	qandadiv, qandaentry, qandaset, question, quote
R	refdescriptor, refentry, refentrytitle, reference, refmeta, refname, refnamediv, refpurpose, refsect1, refsect2, refsect3, refsect4, refsynopsisdiv, remark, row
S	secondary, sect1, sect2, sect3, sect4, see, seealso, set, subscript, subtitle, superscript, synopsis, systemitem
T	table, tbody, term, tertiary, tgroup, thead, tip, title, type
U	ulink, userinput
V	varname
W	warning
X	
Y	
Z	

4.2.5 Troubleshooting DocBook Syntax

The DocBook DTD is strict. If your DocBook documents are not valid, you will get long error messages from the Caché XML (SAX) validator. Fortunately, these error messages include a line number and column offset within the original source file to help you to find the source of the problem.

4.2.5.1 Interpreting Error Messages from the Parser

To illustrate the kind of information generated, suppose we modify a valid XML document by inserting the tag `<AnErroneousTag SomeAttribute="AValue">` just prior to a closing `</para>` tag, with each of these modified tags now appearing on a line by itself:

```
<para>
I have quite a lot to say.
<AnErroneousTag SomeAttribute="AValue">
</para>
```

The attempt to load this document into the database fails and generates the following error messages. Each error message actually appears as a single line of output, but this display introduces line breaks to improve readability:

```
SAX Error (a document could not be loaded):
Unknown element 'AnErroneousTag'
in A:\Parser\Example\source.xml at line 357 offset 16
SAX Error (a document could not be loaded):
Attribute 'SomeAttribute' is not declared for element 'AnErroneousTag'
in A:\Parser\Example\source.xml at line 357 offset 31
DocBook Parser Assertion Failure:
Unsupported tag: <AnErroneousTag>
at line 357 offset 40
SAX Fatal Error (a document could not be loaded):
Expected end of tag 'AnErroneousTag'
in A:\Parser\Example\source.xml at line 359 offset 7
ERROR #6301: SAX XML Parser Error:
Expected end of tag 'AnErroneousTag'
in A:\Parser\Example\source.xml at line 359 offset 7
```

This single error has generated multiple messages to the log. This is not uncommon. The format of an XML document is strictly specified by its document type description, in this case, the DocBook DTD. An error that invalidates the document often runs afoul of multiple constraints in the DTD. Each of these is reported, along with descriptive information about the error and its location, in order of occurrence. The previous set of messages indicates the following problems:

- The tag is not part of the DocBook DTD. The tag name is known and can be processed when the whitespace between the tag name and the attribute name is reached.
- Because the tag is not known, no attribute names are defined for it in the DTD. The parser recognizes this once it completes scanning of the attribute name.
- Once the tag and all its attribute values are scanned, that data is passed to the DocBookParser instance which reports that this is not one of the tags it recognizes.
- The final two lines are generated when the scanner collects the `</para>` and determines that the opening element, `<AnErroneousTag SomeAttribute="AValue">`, is not properly closed before another closing element is found.

Remember that the location reported in the error message is the place where the error is detected. This may not be the place where the error actually occurred. Sometimes a missing end-element tag is not detected until much later in the parsing operation, because the markup which would have followed the closing element could also logically be permitted within the element itself. A working knowledge of the DocBook DTD and the intended structure of the document are indispensable to quickly identify this class of mistakes.

4.2.5.2 Enabling the DocBookParser Trace Capability

The DocBook.DocBookParser class provide trace information to the Terminal log about what happens during the analysis of the document. To enable tracing, set the global `^DocBook.Config("TRACE")` to a nonzero value in the DOCBOOK namespace. For example:

```
USER>znspace "DOCBOOK"

DOCBOOK>set ^DocBook.Config("TRACE")=23

DOCBOOK>zwrite ^DocBook.Config("TRACE")
^DocBook.Config("TRACE")=23

DOCBOOK>
```

The value is interpreted as a bit mask indicating the actions for which actions to produce trace output. The bit values and their associated actions are:

Bit Value	Associated Action
1	Trace the occurrence of start elements; that is, <StartElement>.
2	Trace the occurrence of end elements; that is, </StartElement>.
4	Produce information when an entity is recognized.
8	Indicate when the parser has finished scanning an entity.
16	Display markup and text information which is ignored during the parse.
32	Show the text scanned for block element content.
64	Display progress bars during certain internal operations.

Thus, a value of 23 for `^DocBook.Config("TRACE")` indicates 1+2+4+16=23. That means to trace output for beginning (1) and ending (2) elements, for entities found (4), and for any material in the document which is ignored (16).

5

Legacy Documentation

InterSystems maintains an archive of some material from previous Caché releases. This archive provides a repository for legacy documentation content that either is deprecated or has been superseded. The archive consists of a set of PDF files. You can download them from the InterSystems Web site by clicking the links in this chapter.

Important: This is important.

When using material from the archive, keep the following points in mind:

- The selections available in the archive result from frequent past requests for certain information. Items can be added or removed without notice.
- This material is offered on an "as is" basis. Beyond making the material accessible online, no other editing of the text has occurred.
- InterSystems offers no assurances regarding the completeness or accuracy of the archive, its coverage, or currency; and none should be inferred.
- If the archive content conflicts with current documentation, it is likely that the archive content is outdated and describes functionality that has been updated or superseded.

The current archive contents are:

- *New Locations of Caché 5.0 Advanced Configuration Settings*, Version 2008.2.3
- *Caché Programming Guide*, Version 4.1: "Chapter 9 — Additional Routine Utilities" includes:
  - **%RCHANGE**, **%RCMP**, **%RCMPSEQ**, **%RCOMPIL**, **%RCOPY**
  - **%RD**, **%RDEL**, **%RDELETE**
  - **%RFAND**, **%RFIND**, **%RFIRST**
  - **%RI**, **%RIMF**

  - **%RKILL**
  - **%RO**, **%ROMF**
  - **%RPURGE**
  - **%RVERMAX**

- *Caché Programming Guide*, Version 4.1: "Chapter 10 — Additional Global Utilities" includes:
  - **%G**
  - **%GCHANGE**, **%GCOPY**
  - **%GD**
  - **%GI**, **%GIF**, **%GIGEN**
  - **%GO**, **%GOF**, **%GOGEN**
  - **%GSET**

- *Caché Programming Guide*, Version 4.1: "Chapter 11 — Date, Time, and Numeric Utilities" includes:
  - Date functionality: **%D**, *%DAT* (variable), **%DATE** (including sliding window support), **%DO**, *%DS* (variable)
  - Time functionality: **%T**, **%TI**, **%TIME**, and **%TO**
  - **%NLS** and locales
  - Numeric functionality: **%DOCTAL**, **%DX**, **%OD**, **%SQROOT**, and **%XD**

- *Caché Programming Guide*, Version 4.1: "Chapter 12 — Additional Programmer Utilities" includes:
  - **%CD**
  - **%DIR**
  - **%GLO**, **%GSET**
  - **%NSP**
  - **%PRIO**
  - **%RSET**, **%RSETN**

- *Caché Programming Guide*, Version 4.1: "Appendix A — Routine Development in Programmer Mode"
- *Caché Programming Guide*, Version 4.1: "Appendix C — Physical Storage of Globals and Collation"

- *Object Utility Library Reference* (^**%apiOBJ** routines), Version 3.1
- *Caché Programming Guide*, Version 2.1: "Chapter 9 — Routine Debugging" includes:
  - **%ETN**
  - **%ERN**
- F-DBMS Release Notes:
  - F-11
  - F-12
  - F-13
  - F-14
  - F-15
  - F-17
- Weblink 4.2 documentation:
  - *WebLink Guide*
  - *WebLink Developer Guide*
- *Open M with SQL Data Dictionary Guide* for F.10
- *Open M/SQL Developer Guide* for F.6 and F.7