The parsing process consists of three fundamental steps:

- The z/OS XML parser is an integrated parser for z/OS. There is no need to download or install any additional packages to use the z/OS XML parser.
- The z/OS XML parser provides C/C++ and assembler interfaces for callers to use.
- A caller can access these services through the z/OS XML System Services APIs.

The z/OS XML parser provides a buffer-in, buffer-out processing model instead of the event driven model common to SAX parsers.

The source document and the parsed data stream may actually each span one or more buffers, allowing the caller to feed the document to the z/OS XML parser in multiple pieces as well as receive output from the parser in the same fashion.

- The z/OS XML parser uses buffer spanning to handle documents of unbounded length.
- Buffer spanning enables the z/OS XML parser to use multiple buffers to contain the document being parsed, along with the parsed data stream generated from it.

The z/OS XML parser has minimal linkage overhead.

- The z/OS XML parser provides assistive aids to the user in debugging not well formed documents.
- The caller may need to provide multiple buffers to contain the data stream that the z/OS XML parser traverses through.

The z/OS XML parser supports several code pages.

- The caller must supply the CCSID of the encoding for the document at the time the z/OS XML parser is initialized.

The following steps summarize parsing XML documents using the z/OS XML parser:

1. Call the initialization service. This establishes the PIMA, which is then used to create and manage the PIMA and provide the parser with the initial data structures required to begin the parse process.

2. Call the parse service to parse the document.

3. The application processes the output buffer.

4. Determine if there are additional documents to be processed. If so, call the termination service to terminate the parse process, and repeat Steps 1-3.

**Rule 1:** A particular PIMA must only be used during the parse of a single XML document at a time. Only one PIMA can be reused for the parse of another document. In addition, the PIMA is used as a memory area storing temporary data during the parse.

**Rule 2:** The minimum size for the PIMA is 128K bytes. Everything that the z/OS XML parser uses to handle documents of unbounded length.

**Input Buffer**

- The caller may need to provide multiple buffers to contain the data stream that the z/OS XML parser traverses through.

- Similarly, the caller may need to provide multiple buffers to contain the data stream that the z/OS XML parser generates.

**Output Buffer**

- Document processing is the creation of the output buffers from the parsed input data. As the z/OS XML parser traverses through the input buffer, the output buffer is built.

The left diagram displays the processing model using buffer spanning. It shows both the input and output buffers, where buffers 2-5 represent the additional buffers created to support a large document.

**Usage Considerations:**

- z/OS XML System Services supports several code pages.

- The caller must supply the CCSID of the encoding for the document at the time the z/OS XML parser is initialized.

- For callers that do not provide a memory allocation exit, the z/OS XML parser provides default routines to allocate and free memory.

- The z/OS XML parser also provides an option at initialization time allowing the caller to specify how the z/OS XML parser’s default routine allocates memory.