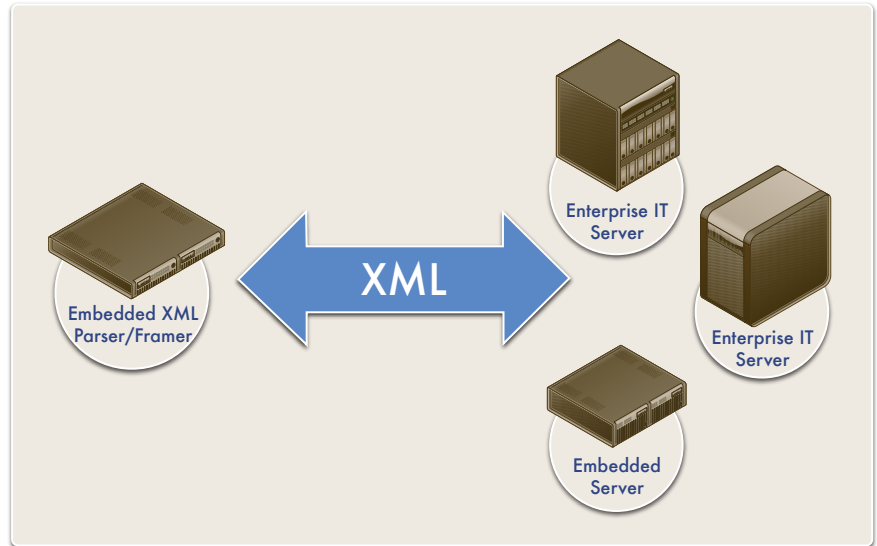


### RomXML AE Benefits

- Decrease time to market by leveraging proven embedded XML technology
- Lightweight Parsing and Framing technology
- Translate XML syntax to and from embedded internal C-language structures
- Ability to create and read XML-based documents without carrying the overhead of general-purpose XML tools
- Utilizes RxSchema to define both C-language internal storage format and the set of XML elements and attributes for XML-based data interchange
- RomXML AE TagBuilder utility program drastically reduces coding effort by analyzing object definitions and produces C-language source with structures pre-defined and ready to use with Parsing and Framing functions
- Small RAM/ROM footprint
- ANSI-C Implementation
- Compilation switches for size and speed trade-offs
- Ability to use standalone or in conjunction with Allegro's RomPager AE suite of products
- Highly portable



XML offers the exciting potential for a true cross-platform data exchange format. Allegro's RomXML AE Parsing and Framing Toolkit enables your embedded development team to rapidly build and deploy embedded robust XML-enabled applications. Originally built from the start for use in demanding embedded environments, RomXML AE provides a lightweight technology to create (frame) and read (parse) XML-based documents without carrying the overhead of general-purpose XML-based solutions.

### Data Portability, XML, and Embedded Systems

Data portability has long been an issue for the computing industry at large. A readable document or file on one platform is not necessarily readable on another or with another application on the same platform. Worst yet, it is unlikely to be readable with a future or past version of the same application on the same platform. Depending on the complexity of the data, the validity from one application and platform to the next is also uncertain. A historic example that illustrates this point is the volumes of data generated by the embedded systems onboard the original Gemini and Apollo space missions in the late 1960s and early 1970s. Historic flight data acquired from these missions and written in a binary format to tape is likely lost forever. Like the fast-paced embedded development environments of today, recovering data from previous generations of projects or products combined with the effects rapid change in hardware technology in addition to changes in staff provide a significant challenge. XML offers an extensible human and machine readable format that significantly eases the burden of data portability between computing platforms as well as through time.

Many embedded systems today are increasingly members of the connected and inter-connected world of data producers and data consumers. When connecting data sources to create a larger enterprise system corporate IT managers see the explicit value of data portability across heterogeneous platforms especially embedded systems. As an example, it is common to find embedded systems in automobiles producing real-time data in XML for consumption by corporate IT centers that enable previously unheard levels of customer service. Others examples include the exchange of data between IT servers and embedded handsets to display directory information, stock ticker prices and the latest breaking news.

## XML Parsing and Framing

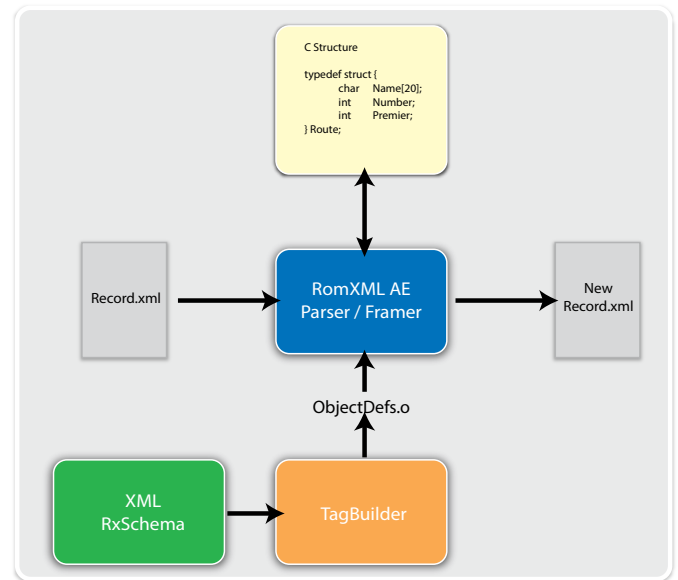
Within enterprise environments, there are two common methods employed for dealing with XML. One approach is a method called Simple API for XML (SAX). In the SAX approach, as a document is parsed, XML elements and attributes are passed to a user program for interpretation. The XML information and underlying structure are discarded in this parsing method once the data is passed to the user program. While this method doesn't use much memory, it requires the user program to deal with XML in a sequential manner rather than as a whole entity. Since the XML syntax and underlying structure are discarded, SAX style parsers do not provide much support for the framing process.

The second approach typically utilized in an enterprise environment is a method called the Document Object Model (DOM). A DOM parser recognizes that an XML document is a hierarchical data format and stores a tree structure representing both the underlying XML information and data. An API is provided to traverse the tree structure in any order and allows access to the entire XML document. Framing a new document is very straight forward since the XML data and overall structure are retained. However, the DOM method for parsing and framing documents is very memory intensive often utilizing over twice the memory of the original document.

The RomXML AE toolkit is designed for embedded devices that have limited resources and wish to use a more powerful method than SAX, without the overhead of DOM. For embedded applications, devices do not need the ability to parse and frame general purpose XML documents. However, embedded devices do require the ability to efficiently translate device specific data between an internal format (such as a C structure) and XML syntax. This means the schema (XML element definitions) for a given document do not change and can be specified at compile time. The RomXML AE toolkit provides a light weight translation between pre-defined C-Language structures and XML formatted representations. This enables an embedded device to exchange data as well-formed XML documents without the overhead of general-purpose XML translators.

### TagBuilder

As part of the RomXML AE toolkit, the TagBuilder compiler further decreases your development effort when utilizing XML in your embedded design. RomXML AE uses a special set of tags (RxSchema) to define an XML object. The RxSchema language allows an XML object to be defined with both C-language internal storage format and the set of XML elements and attributes used with XML-based data exchange. The RomXML AE TagBuilder compiler analyzes the RomXML AE RxSchema object definitions and produces an object definition file in C-language source that is compiled with the RomXML AE Parser and Framer code. The definition file contains the transfer tables that the runtime RomXML AE code utilizes to perform specific translations.



### Flexible Design

The RomXML AE runtime is delivered in ANSI-C, is highly portable and offers a rich API for your development team to handle XML objects. RomXML AE can handle XML datastreams transmitted with your proprietary communications methodologies or in connection with Allegro's RomPager AE suite of products. When used together with the rich RomPager AE suite, web services such as SOAP and XML-RPC are also available for use in your design.