

# SOLA DOM API Overview



**Revision Date: August 2017**



## Table of Contents

|  |    |
|--|----|
| Table of Contents.....                 | ii |
| The SOLA DOM API .....                 | 3  |
| SOLA DOM API Functions .....           | 5  |
| Data Conversion Utility XMLPC107 ..... | 11 |



## The SOLA DOM API

The Document Object Model (DOM) is a specification designed by the World-Wide-Web Consortium (W3C) to provide an object oriented and vendor independent way to inquire on and modify XML documents. DOM works by loading the entire XML document into memory and then converting the document into a tree structure. Inquiring on, modifying and creating a new XML document is done using methods provided by the DOM specification.

SOLA provides the SOLA DOM API to inquire on, modify and create XML documents. The SOLA DOM API is a "DOM like" API that implements the functionality of DOM in a format that is easily used by procedural languages like COBOL. The SOLA DOM API can be used by CICS transactions and batch jobs. It provides functions to inquire on an XML document repeatedly in any direction and modify an existing document. The API also provides methods to create new XML documents. This document lists all the features of the SOLA DOM API and provides many samples for use in mainframe programs running in CICS and batch.

In most cases, the SOLA DOM API is identical for CICS and batch programs. In those instances where the APIs differ, the differences will be noted in the text.

### **Advantages**

The SOLA DOM API provides an easy and efficient way for any mainframe program to process XML documents. Performance (CPU utilization and memory consumption) was a major focus in the design and development of the SOLA DOM API. To keep CPU consumption to a minimum, the SOLA DOM API is highly optimized and is written entirely in Assembler for z/OS. The SOLA DOM API has a relatively small memory footprint for the DOM tree due to the fact that it parses the XML document and normalizes it in place without requiring any additional memory to hold the node values. The DOM tree contains pointers to the parsed and normalized document, thereby holding the memory footprint to a minimum.

The SOLA DOM API makes it easy for the mainframe programmer to produce well-formed XML without the need to perform complex string manipulations. Without the SOLA DOM API, the programmer would have to follow the XML



specifications provided by the W3C which can be a time consuming and daunting task.

For example, the following table shows incorrect XML formations that will be rejected by an XML parser:

|   |  |
|---|--|
| <code>&lt;SSN#&gt;212-99-9654&lt;/SSN#&gt;</code>                   | The element name can't contain<br>"#" Character in XML version 1.0 |
| <code>&lt;SecurityName&gt;AT&amp;T<br/>&lt;/SecurityName&gt;</code> | The special entity "&" should be<br>represented as &amp;           |

Because the SOLA DOM API conforms to the W3C specifications, XML documents created using the SOLA DOM API will always be valid and well formed.



## SOLA DOM API FUNCTIONS

The SOLA DOM API is an assembler program callable from any mainframe program. The API supports inquiry and modification of XML documents as well as the creation of new XML documents.

### XML Inquiry Functions

|                                |  |
|--------------------------------|--|
| <b>getAttribute</b>            | <p>This function retrieves the value for a given attribute name in a name/value pair.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |
| <b>getAttributeAddress</b>     | <p>This function is identical to <code>getAttribute</code>, except that the attribute address is returned in place of the attribute value.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>   |
| <b>getAttributeAddressById</b> | <p>The <code>getAttributeAddressById</code> function is identical to <code>getAttributeById</code>, except that the attribute value address is returned in place of the attribute value.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p> |
| <b>getAttributeArray</b>       | <p>This function retrieves an array containing all the attributes for a given element node.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |
| <b>getAttributeById</b>        | <p>This function retrieves an attribute and its value using a given node Id.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>   |
| <b>getElementAddress</b>       | <p>This function is identical to <code>getElementByTagName</code>, except that the element value address is returned in place of the element value.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>                                      |



|                              |  |
|------------------------------|--|
| <b>getElementAddressById</b> | <p>The getElementAddressById function is identical to getElementById, except that the element value address is returned in place of the element value.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>   |
| <b>getElementById</b>        | <p>This function retrieves an element and its child text node using a given node Id.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>   |
| <b>getElementByTagName</b>   | <p>The getElementByTagName retrieves an element node by its tag name.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |
| <b>getElementByXPath</b>     | <p>This function retrieves an element and its child text node for a given XPath expression.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |
| <b>getNamespace</b>          | <p>This function retrieves the value of a namespace associated with a namespace unique node identifier.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |
| <b>Parse</b>                 | <p>This function parses, normalizes and creates the tree structure for a given input XML document.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>   |
| <b>resetSelectAttrNodes</b>  | <p>This function is used to reset the internal counters used by the selectAttrNodes function. The function should only be used in conjunction with the selectAttrNodes function when it is operating in an iterative mode.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p> |



|                               |  |
|-------------------------------|--|
| <b>resetSelectNodes</b>       | <p>This function is used to reset the internal counters used by the selectNodes function. The function should only be used in conjunction with the selectNodes function when it is operating in an iterative mode.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p> |
| <b>selectAttrNodes</b>        | <p>This function is used to inquire on either the total number of attribute nodes in an XML document, to get a particular attribute in the sequence or retrieve attributes iteratively.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>                            |
| <b>selectAttrNodesAddress</b> | <p>This function is identical to selectAttrNodes, except that the address of the attribute value is returned instead of the value of the attribute.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |
| <b>selectNodes</b>            | <p>This function is used to inquire on either the total number of element nodes in an XML document, to get a particular element in the sequence or retrieve elements iteratively.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>                                  |
| <b>selectNodesAddress</b>     | <p>This function is identical to selectNodes, except that the address of the element is returned instead of the value of the element.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |

**XML CREATION AND MODIFICATION FUNCTIONS**

|                          |  |
|--------------------------|--|
| <b>appendChild</b>       | <p>This function appends a new child element to a specified parent element. Optionally, a text node value can be attached to the element.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |
| <b>appendChildBefore</b> | <p>This function is used to add an element before a particular element. The element node to be added will share the same parent as the element that it is to be appended before.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p> |
| <b>appendChildNL</b>     | <p>This function is identical to <code>appendChild</code>, except that it will also insert a newline character into the document before appending the child element node.</p> <p>This is a <b>batch</b> only function.</p>                             |
| <b>appendTextNode</b>    | <p>This function is used to add a child text node to an element node.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |
| <b>createDocument</b>    | <p>This function creates the root element of an XML document.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |





|                     |  |
|---------------------|--|
| <b>finalize</b>     | <p>The finalize function is used to finalize and optionally retrieve a newly created XML document. Finalize will convert the DOM tree into an XML stream, as it does so all open tags will be closed and the document will be made ready to be externalized. The finalize function returns a pointer to the XML document.</p> <p>Once finalized, the document can be accessed with the address pointer. Alternatively, the document can be accessed using the retrieve function.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p> |
| <b>getXMLLength</b> | <p>This function retrieves the length of the completed XML document.</p> <p>This is a <b>batch</b> only function.</p>  |
| <b>removeNode</b>   | <p>This function removes any type of node (element, attribute or namespace node) from an XML document. To remove text nodes, use the updateTextNode function.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |
| <b>retrieve</b>     | <p>This function finalizes the document and then gets the complete (if appendChildNL was not used) document and copies it to a data area that you provide. The SOLA DOM API will then release any storage space used by the DOM tree and internal control block.</p> <p>If appendChildNL was used then the retrieve function will fetch the document in pieces, each piece ending in a newline.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |
| <b>setAttribute</b> | <p>This function is used to add and attach attributes (name/value pairs) to an element.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>  |



|                        |   |
|------------------------|---|
| <b>setNamespace</b>    | <p>This function is used to add and attach a namespace to an element.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>   |
| <b>updateAttribute</b> | <p>This function is used to update the text of an existing attribute. Changing the attribute name is not allowed (to change the name of an attribute you must first remove and then set the attribute).</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p> |
| <b>updateTextNode</b>  | <p>This function is used to change the value of or to remove an existing text Node under an element node.</p> <p>This function can be used in <b>CICS</b> and <b>batch</b>.</p>   |



## Miscellaneous Functions

|                    |   |
|--------------------|---|
| <b>freeStorage</b> | This function frees any storage that has been acquired by the DOM parser and SOLA DOM API.<br><br>This function can be used in <b>CICS</b> and <b>batch</b> . |
|--------------------|---|

## DATA CONVERSION UTILITY XMLPC107

SOLA provides a utility program, XMLPC107, to provide data conversion between native mainframe formats and XML. The utility is an assembler program that is implemented with a callable interface.

Although the utility provides functions for converting character strings, it should not be used for these conversions if you are using the SOLA DOM API. This is because the SOLA DOM API uses the utility internally for XML encoding and the effect of encoding a string twice is unpredictable.