Attachments in SOAP Messages

An Oracle White Paper
December 2005

ORACLE

Attachments in SOAP messages

EXECUTIVE SUMMARY

With more Web Services relying upon attachments to meet the business needs for sending extra business data related to the SOAP messages, choosing the right technology for using attachments in SOAP messages among ever evolving standards is possible by understanding those standards from the viewpoint of processing model, and by evaluating each of them against the requirements.

INTRODUCTION

Packaging as attachments in SOAP messages has become a norm in the Web Services area for any data that cannot be placed inside SOAP Envelope. In the presence of several competing standards on utilizing attachments in SOAP messages, there has been tremendous confusion about which standard to choose for scenarios with different sets of requirements. This paper describes the background for leveraging attachments in SOAP messaging, and presents a brief history and an outline of each standard. It also details the impact of attachments on security, interoperability, and performance, and lays down the guidelines on how to pick a right technology in the conclusion.

SOAP AND ATTACHMENTS

SOAP Basics

SOAP in Web Services

Service Oriented Architecture (SOA) has been the most recent trend in building distributed enterprise applications in an open, loosely-coupled way, and it is primarily through Web Services that the visions and promises of SOA are realized in its fullness. Standards and technologies around Web Services have been evolving past the earlier infant stages and have garnered wide acceptance from the industry. At the center of Web Services is SOAP at the messaging layer that connects Services Providers and Services Consumers potentially via multiple Intermediaries that can process the SOAP messages.

SOAP 101

SOAP is an XML-based messaging protocol. It started out as a cross-platform, language-neutral Remote Procedure Call (RPC) mechanism, replacing CORBA or other distributed computing technologies. Later on, it evolved into a generic messaging mechanism that accommodates various Message Exchange Patterns (MEP) including two-way request-

response pattern for Remote Method Invocation (RMI) and one-way posting pattern for transferring complicated documents such as purchase order requests.

How to package SOAP messages on the transport wire and how to interpret them at the endpoint are dictated by WSDL. Based on the syntactical and semantical definitions given in WSDL, a SOAP Body and possibly zero or multiple SOAP Headers get packaged into the SOAP Envelope. WSDL also spells out transport binding, message encoding styles, and Services endpoints.

With the lessons learned from SOAP1.1, SOAP1.2 takes the standard to the next level by clarifying SOAP Header processing model, making SOAP Fault more useful, and introducing the application/XML+SOAP media type to differentiate SOAP messages from ordinary XML documents.

Attachments in SOAP Messages

Motivation

Limitations of embedded SOAP messages

The idea of embedding every SOAP Element in SOAP Envelope has shown its limitations on the payload type and size. Certain data types are not just XML-friendly and cannot reside inside the XML document that any SOAP Envelope is. Binary data in eight bits, for instance, should be converted into base64binary type in XML Schema Definition (XSD) to become a valid SOAP Element, and, after the conversion, all the associated metadata gets lost. Furthermore, conventional XML parser cannot parse very well XML documents that are deep or complex in structure, or those that are simply huge in size.

When it comes to binary data or XML documents that do not need to be unmarshalled to objects, a possible, but painful and inelegant, solution would be to send them in separate channels and to keep track of their relationship to the SOAP Envelope. For instance, when sending the product image with related purchase order XML document, one can use SOAP for order summary, and FTP for image and SMTP for XML files. In this case, the cohesive relationship between the order entry and the product image file or purchase order document can get broken.

Marriage with MIME

In an attempt to address the issues around embedded SOAP messages, the idea of using MIME packaging was proposed as a scheme to put data outside the SOAP Envelope as attachments. MIME was originally devised for emails to allow one to send arbitrary attachments together with the email message, and HTTP borrows some concepts from MIME mainly for the purpose of specifying acceptable content types in HTTP response. MIME Encapsulation of Aggregate Documents such as HTML (MHTML) is another example of using MIME and CONTENT-ID URIs as a way to link document entities external to the root document.

Since MIME allows eight bit binary data, there is no need to perform binary64 conversion, which not only takes extra time for encoding and decoding, but also makes the payload 30%

bigger on the average. With MIME, virtually the limit to the data type and the number of entries has been cleared up. MIME is also streaming-friendly in nature, helping to manipulate bulky payloads more smoothly.

Series of standards

SOAP Messages with Attachments (SwA)

SwA applies MIME attachments to SOAP, using multipart/mime content type, putting the SOAP Envelope in the root MIME part and other related attachments in ensuing MIME parts inside the MIME package. It relies on HREF attribute and Content-ID MIME header to relate attachments to SOAP message parts.

Being the first manifestation of the notion of adopting MIME in the context of SOAP, SwA is not the optimal technique in spite of the premier step it took. After all, SOAP messages are more structured than email messages, and require tighter reference schemes. SwA is more popular in the Java camp than the rest of the world. Besides, B2B standards such as ebXML use SwA as the messaging layer for its flexibility.

Direct Internet Message Encapsulation (DIME)

MIME has fundamental flaws in design. Mime part boundary matching takes computing time proportional to the size of the MIME message. More fatally, although the chances are practically low, it is possible to have a false boundary if the content has the same pattern as the MIME part boundary string by accident.

Those problems have been known even before the idea of using MIME for SwA was tossed out. As an alternative to MIME, DIME was proposed as a generic messaging mechanism that depends on offset values instead of string-based boundaries. A DIME message is composed of one or more DIME records in a binary message format that can encapsulate multiple application-defined payloads of arbitrary type and size into a single message construct.

WS-Attachments

Being one of numerous WS-* specifications, WS-Attachments lays out how to apply DIME in the Web Services Context. The fact that shortcomings of SwA turned out to be not as severe as pointed out by DIME proponents made it hard to convince the industry to desert SwA and adopt DIME-based WS-Attachments. Due to the lukewarm responses, it has been officially declared to be obsolete.

Proposed Infoset Addendum to SOAP Messages with Attachments (PASwA)

Both SwA and DIME (or WS-Attachments) introduce data structures that are external to XML Infoset and it is not well defined how to relate attachments to the SOAP Envelope. PASwA is the first fruit out of the efforts to introduce the attachment processing model solely based on XML Infoset for the purpose of providing a consistent logical view of SOAP messages regardless of existence of attachments. It defines a set of constructs that help to describe relationships between attachments and SOAP messages.

XML-binary Optimized Packaging (XOP)

W3C embarked on elaboration of the concepts of PASwA in separate W3C specifications. XOP defines how to serialize SOAP messages with binary contents, preserving the XML Infoset. It is more generic than PASwA in that XOP can be used for generalized packaging models including MIME. In other words, XOP is applicable to SOAP, but not limited to it.

SOAP Message Transmission Optimization Mechanism (MTOM)

MTOM, a part of SOAP 1.2 specification, applies XOP to SwA. Staying in the XML Infoset, MTOM-based attachments are equivalent to embedded SOAP elements semantically to the endpoint SOAP Nodes. With many Web Services standards defined on XML Infoset model, its significance lies in the fact that the presence of attachments in SOAP messages is no longer an exceptional or special case.

MTOM messages are valid SwA messages on the network although the runtime should accept and understand XOP/MIME content-type to be able to recover the XML Infoset out of attachments, following the processing model in the specification. Many SOAP vendors are expected to release MTOM-enabled SOAP toolkits and runtime near in the future.

Mechanism

WSDL standard provides several hooks for extensibility for SOAP or Services invocation in general. Being like Java interfaces, PortType section is completely attachment-agnostic. All the standards described above define Binding extension in WSDL as an indication of using respective technology, WSDL1.1 Mime Binding being one for SwA. Later standards rely heavily on XML Schema Definitions in order to capture properties of attachments and to define links between SOAP Elements and attachments. In addition, each standard defines the message formats on the wire with new MIME headers and how to compose and process those messages.

OTHER CONSIDERATIONS

Relationships with other standards

WS-Security

WS-Security is a standard whereby SOAP messages can be sent securely even over an insecure network. Although one can use, for example, SSL to have a secure transport, it cannot support the message-level security. WS-Security not only enables sender authentication through various credential tokens including username/password, X.509 certificates, or SAML token, but also provides data confidentiality through XML Encryption, and data integrity through XML Digital Signature.

WS-Security, being a standard built on XML Infoset, cannot associate encrypted or signed attachments with SOAP envelope in case of SwA or DIME messages. For that reason, as of today, SOAP Requester should encrypt or sign sensitive payload manually before sending as attachments via SwA or DIME, and subsequently, SOAP Provider has to carry out decryption or signature validation at the Service implementation layer. On the contrary, WS-Security can

go along well with MTOM because attachments created and processed via MTOM are no different from SOAP elements from the XML Infoset point of view.

BPEL

Business Process Execution Language (BPEL) is another key standard that extends the definition of Services in SOA by way of Services composition. Business Process is modeled as a series of Web Services invocations for the purpose of integrating business functionalities available inside or across the organizational boundary and orchestrating them as a long running transaction with limited compensatory measures. BPEL itself extends WSDL for its own purpose--to define PartnerLinks for PortType.

Being neutral on the message formats, BPEL does not dictate whether any Web Service should use attachments or not, nor does it impose usage of any particular attachment technology. As a matter of fact, BPEL itself has no built-in support for attachments in SOAP messages. As a result, invoking Web Services using DIME or SwA attachments in the BPEL process is not supported in many BPEL implementations without resorting to custom development. MTOM can mitigate the pain, if not eliminate it altogether, by making the attachments look like SOAP elements transparently.

Interoperability

Publishing and consuming interoperable Web Services was one of the striking challenges identified in the earlier history of Web Services, and enormous progress has been made through WS-I consortium. However, it is only very recently when interoperability for SOAP messages with attachments started drawing attention.

Among various standards on attachments, only SwA has a Basic Profile defined by WS-I, named WS-I Attachment Profile (AP1.0). AP1.0 not only clarifies major ambiguities in WSDL extension for MIME binding, but also tackles SwA's fundamental limitations by introducing "swaRef" attribute as a logical reference to attachment part from the SOAP Body. Many SOAP vendors have also implemented DIME, but there is no official standard for DIME interoperability.

Streaming

Using attachments alone does not make it any easier handling sizable SOAP messages efficiently. Nonetheless, SOAP message parts bound to attachments are easier to utilize streaming because those attachments can be created directly from the data sources, or vice versa, independently of the SOAP envelope.

Streaming can take place along the transport pipe as well as at the endpoint. Taking advantage of HTTP chunking, for instance, can make it more feasible to pass data of virtually unlimited size. In addition, a streaming-enabled XML parser or SOAP processor consumes only a constant amount of memory to build or process SOAP messages with giga bytes of payload data out of LOB database table column. There is no official standard on streaming SOAP messages yet, and it is left to SOAP implementers' hands on how to support the critical requirement.

Front-end SOAP Gateway

Many SOAP containers in the market are vulnerable to Denial-Of-Service (DoS)-type attacks, and the root cause of the danger lies in the in-memory-based XML parser. Whereas this threat is not unique to SOAP server, and any applications using XML can suffer from similar symptoms caused by exhaustion of physical/virtual memory, SOAP endpoints allowing attachments are easier targets for attack. Placing the SOAP Gateway can be a countermeasure, protecting the backend SOAP container from malicious requests. It can

- Enforce authentication/authorization scheme at the transport level: use SOAP over HTTPS when applicable and possible
- Enforce authentication/authorization scheme at the message level: use WS-Security when applicable and possible
- Filter out suspicious messages: examine the HTTP payload size
- Validate attachment metadata: count the MIME parts for attachment and check the content types of each

Oracle's offering

As the leading SOA development toolkit and runtime environment provider, Oracle has a comprehensive list of technology and solution offerings in Web Services area. To highlight a few,

- JAX-RPC implementation for SOAP 1.1 and SOAP 1.2 is available in AS 10.1.3.
- Oracle supports streaming for SwA in AS 10.1.3. To interoperate with Microsoft .NET, DIME is available.
- Oracle AS 10.1.3 supports WS-Security Username Token Profile, SAML Token Profile as well as Binary Token Profile for X.509 certificate based authentication, XML encryption and digital signing.
- Oracle Web Services Manager (OWSM), as the SOAP Intermediary, can provide policy-based authentication and authorization. It is possible to plug a policy enforcement step that can validate attachment MIME parts.
- Oracle HTTP Server (OHS) can be configured to drop malicious requests (a custom Apache module needs to be written)
- MTOM (and apparently XOP) implementation will be available in the upcoming releases.

USING ATTACHMENTS IN SOAP MESSAGES

What to choose today

Inundated by numerous standards around SOAP attachments, SOA architects and developers need to closely look at strengths and limitations when evaluating each of them. In the absence of "one-size-fits-all" technology, following requirements drive the selection of the most appropriate attachment standard.

For handling huge attachment payload

Streaming capability deserves the most weight in this regard. Whereas it is possible to stream data at the sending or receiving endpoints, typically the transport level streaming should be provided by the SOAP runtime. Nothing else is more instrumental than streaming for higher throughput and lower response time when it comes to managing large attachments.

For securing data

Given that neither SwA nor DIME can provide message-level security for the moment, the only practical option for sending attachments securely is to use a secure transport layer such as SSL unless one opts to encrypt and decrypt data by hand. WS-Security can work only for embedded SOAP messages.

For interoperability among heterogeneous platforms

SwA is the most widely adopted standard on the Java side with well-defined set of APIs. DIME is also commonly available in many SOAP products. Just like everything else, following the proven best practices substantially increases the positive chances. And yet, to achieve the highest level of interoperability, one should consider using embedded packaging through base64encoding unless the content is extraordinarily big.

What to expect tomorrow?

The requirement list for SOA-based applications will continue to grow especially around the capability to send data of various media types and of varying size as attachments. Truly document-oriented Web Services will be the majority of the deployed Services for most enterprise applications, and it only means more Services will make use of attachments.

Although nobody can predict that MTOM would be the last standard we are going to see, the initial response from industry is very promising. Interoperable, streaming-enabled MTOM implementations with WS-Security support will accelerate the adoption of attachments in SOAP messaging.

CONCLUSION

Creating and consuming interoperable SOAP messages with attachments is more and more demanded for various Web Services. The web services development community and industry have come a long way, producing a handful of related standards, learning numerous lessons over the course, and the journey has yet come to an end. For the moment, by fully assessing the necessary requirements and understanding the implications of the choice, it is possible to apply the most effective technology to handle SOAP messages with attachments in a secure, reliable, and scalable fashion.

REFERENCES

Simple Object Access Protocol (SOAP): http://www.w3.org/TR/soap/

Simple Object Access Protocol (SOAP) 1.1: http://www.w3.org/TR/soap11/

Simple Object Access Protocol (SOAP) 1.2: http://www.w3.org/TR/soap12/

Web Services Description Language (WSDL) Version 1.1: http://www.w3.org/TR/wsdl

Web Services Description Language (WSDL) Version 2.0: http://www.w3.org/2002/ws/desc/

Multi-purpose Internet Mail Extension (MIME): ftp://www.ietf.org/rfc/rfc2387.txt

SOAP Messages with Attachments: http://www.w3.org/TR/SOAP-attachments

WS-Attachments: http://msdn.microsoft.com/library/en-us/dnglobspec/html/draft-nielsen-dime-soap-01.txt

DIME: http://msdn.microsoft.com/library/en-us/dnglobspec/html/draft-nielsen-dime-02.txt

Proposed Infoset Addendum to SOAP Messages with Attachments: http://www.gotdotnet.com/team/jeffsch/paswa/paswa61.html

XML-binary Optimized Packaging (XOP): http://www.w3.org/TR/2004/CR-xop10-20040826/

SOAP Message Transmission Optimization Mechanism (MTOM): http://www.w3.org/TR/soap12-mtom/

WS-I Attachment Profile 1.0: http://www.ws-i.org/Profiles/AttachmentsProfile-1.0-2004-08-24.html

WS-Security 1.0: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

Business Process Execution Language (BPEL): ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf

MIME Encapsulation of Aggregate Documents such as HTML (MHTML): http://www.ietf.org/rfc/rfc2557.txt

Security Assertion Markup Language (SAML): http://docs.oasis-open.org/security/saml/v2.0/saml-2.0-os.zip

Attachments in SOAP messages
December 2005

Author: Pyounguk Cho
Contributing Authors:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com