



Implementing the Confirm BOD Version 2.0

for OAGIS[®] Release 9.+

an OAGi Normative Specification

Document #20110501V2.0

Open standards that open markets[™]

Open Applications Group, Incorporated – OAGi[®]
A consortium dedicated to simplifying business across all value and information chains
www.OAGi.org

Copyright 2011 OAGi

Prepared by:

Open Applications Group, Inc.

Architecture Council, 2011

Special Thanks to the following:

Steffen M. Fohn

ADP – Representative

Michael Rowell

OAGIS - Chief Architect

David Connelly

OAGIS - CEO

Notice

The material contained herein is not a license, either expressly or impliedly, to any intellectual property owned or controlled by any of the authors or developers of this material or OAGi. The material contained herein is provided on an "AS IS" basis and to the maximum extent permitted by applicable law, this material is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material and OAGi hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence.

ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, AND CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THIS MATERIAL.

IN NO EVENT WILL ANY AUTHOR OR DEVELOPER OF THIS MATERIAL OR OAGi BE LIABLE TO ANY OTHER PARTY FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THIS OR ANY OTHER AGREEMENT RELATING TO THIS MATERIAL, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

For more information, contact:

Open Applications Group, Inc.

P.O. Box 4897

Marietta, Georgia 30061 USA

Telephone: +1 678 715 7588

Internet: <http://www.openapplications.org>

Contents

81	Contents	
82	Introduction	6
83	Purpose of this Document	6
84	Terminology and Notation.....	7
85	Conformance	7
86	ConfirmBOD Specifications	8
87	Assumptions when using the ConfirmBOD	8
88	ConfirmBOD Purpose and Intent	9
89	Web Service Example.....	11
90	BODs, Verbs Paired with the ConfirmBOD	11
91	How to use the ConfirmBOD.....	12
92	Requesting a ConfirmBOD	13
93	The ConfirmationCode	13
94	Producing a ConfirmBOD	14
95	BOD and Noun Messages in ConfirmBOD.....	15
96	BOD Messages.....	15
97	Noun Messages.....	18
98	Warning and Error Messages.....	20
99	Identifying BOD Instances.....	21
100	Identifying Noun Instances.....	21
101	ConfirmBOD Best Practices (non-Normative).....	22
102	Data Integrity	22
103	Multiple Noun option for OAGIS® BODs.....	23
104	Publish and Subscribe Error Handling	23
105	Centralized Error Management	23
106	Centralized Monitoring	23
107	Data Synchronization Integrity	24
108	Appendix A – Glossary	25
109	Appendix B: Sample System Flows.....	26
110	B.1 Scenario 1	26
111	B.2 Scenario 2.....	26
112	Appendix C: Success and Error Scenarios.....	28
113	C.1 Scenario 1.....	28
114	C.2 Scenario 2.....	28

115	C.3 Scenario 3.....	29
116	C.4 Scenario 4.....	29
117		

Introduction

Purpose of this Document

This document is a normative (standard) guide to implementing the ConfirmBOD in OAGIS® Standard based implementations. The ConfirmBOD is a generic BOD that serves as a general response message for processing status information. The ConfirmBOD facilitates the return of message processing success, warning and error for all messages and processes.

This document is not meant to address the entire issue of business transaction confirmation. That issue is addressed by the normative document describing the use of OAGIS® Verbs in a business process, business transaction, or business data exchange for the purpose of business software interoperability.

The ConfirmBOD is one of many OAGIS® Standard messages; there is often debate by OAGIS® users as to when and how to use the ConfirmBOD, particularly in business scenarios.

Because the ConfirmBOD is usable in practically all OAGIS® scenarios, this BOD is a special case for guidelines for deploying it.

[Note] The ConfirmBOD is not meant to replace or compete with transport frameworks such as SOAP, ebMS or RNIF (RosettaNet Implementation Framework). It is a business processing message that is described in much more detail throughout this document.

This document defines the original intent of the ConfirmBOD, when to use the ConfirmBOD, and also some examples of using the ConfirmBOD.

The document is subject to revision from time to time, so it is recommended that you check back to the Open Applications Group (OAGi) web site to ensure that you have the most recent version of this document. www.openapplications.org.

This document specification replaces the OAGi document, "Using the ConfirmBOD, A Normative Guide to Implementing the ConfirmBOD v1.0" [Connelly, Rowell] (2005)¹.

¹ Using the ConfirmBOD: A Normative Guide to Implementation the ConfirmBOD v1.0 [Connelly, Rowell] (2005).

Terminology and Notation

The key words, “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in Internet Engineering Task Force (IETF) Request For Comments (RFC) 2119.²

Where ever *xsd:* appears it is references to constructs from W3C XML schema specification.

The following are notations that are used throughout this document:

- Example – A representation of a definition or rule that are intended to be informative.
- [Note] – Explanatory information that is intended to be informative.
- [OAGi R n] – Identifies a rule n that is specified by this document that requires conformance.
- [Note] Rules are normative. In order to ensure continuity across versions of the specification, rule numbers that are deleted will not be re-issued and any new rules will be assigned the next higher number regardless of the location.
- When defining rules the following annotations are used:
 - [] – Optional
 - < > - Variable
 - | - Choice
 - Italics are used to highlight or emphasize a term or phrase

Conformance

Applications will be considered to be in full conformance with this technical standard if they comply with the content of the normative sections, rules and definitions.

² Key words for use in RFCs to Indicate Requirement Levels. Internet Engineering Task Force, Request for Comments: 2119, March, 1997, <http://www.ietf.org/rfc/rfc2119.txt>

ConfirmBOD Specifications

Assumptions when using the ConfirmBOD

The OAGi model for building OAGIS® addresses the business application integration problem from a business, top down point of view. This is central to understanding how to use and deploy the OAGIS® standard.

The business application integration problem is faced by enterprises that must manage an application portfolio. The enterprise needs to assemble, and re-assemble, their business applications in order to connect to their customers and their suppliers, as well as to meet the changing needs of their business inside the enterprise as well. See the figure 1 for a graphical representation of this.

The OAGi model assumes that the middleware and network layers, including reliable messaging and security, if necessary, are available to be leveraged.

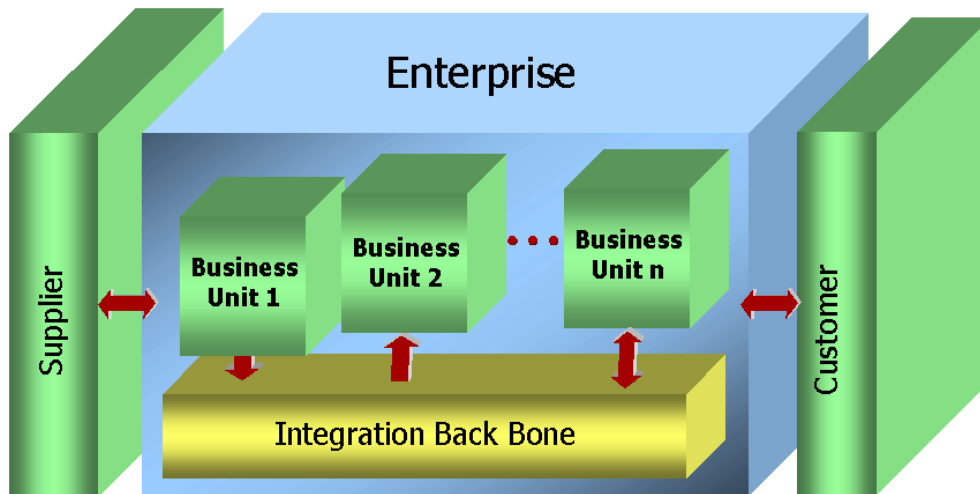


Figure 1 – Configurable Service Oriented Architecture for the Enterprise.

Figure 2 shows that the ConfirmBOD is intended for Application-to-Application or Business-to-Business communication of receiving and understanding the originating BOD.

Using the ConfirmBOD

NOTE: Parser and constraint processing may reside either in the middleware or the business application.

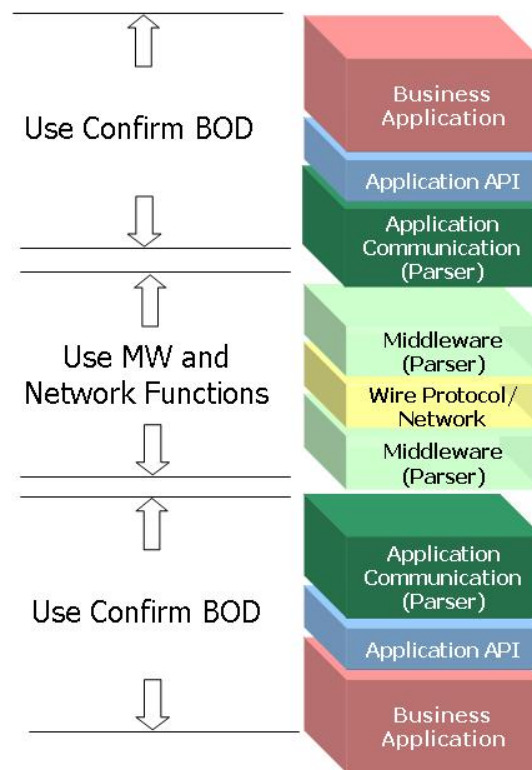


Figure 2 – Use of the ConfirmBOD for Application and Business-to-Business communication.

The purpose of the ConfirmBOD is to function at the application layer of the communication stack that provides for business application to business application exception processing within a business process, business transaction, or business data exchange for the purpose of business software interoperability.

ConfirmBOD Purpose and Intent

The American Heritage dictionary defines “Confirm” as “to support or establish the certainty or validity of”, and even more specifically as, “the act of establishing the certainty or validity”. Consistent with this meaning, a message sender may ask the receiver to confirm processing of a message. This request is designated with a specific set of codes in a BOD instance’s ApplicationArea. The response to the confirm request may indicate the success of processing a BOD instance, the failure of processing a BOD instance, or the partial failure of processing a BOD instance. OAGi named the response message to a confirm request as ConfirmBOD.

The ConfirmBOD is to be used for confirmation of transactions for business applications. It is not meant to replace Messaging Service Confirmation or to replace TCP/IP or similar protocol confirmation.

[Note] This is not dependant on the physical implementation of the integration architecture. It is dependant on the type of processing to be done, that is business transactions, not transport protocol transactions.

Consider, for example, a parser that implements business rules on the data or the process, then that parser is involved in the business transaction processing and must be able to issue the ConfirmBOD in order to be OAGIS® compliant.

Figure 3 is a simplified picture of exception processing within a business process, transaction, or data exchange for the purpose of business application interoperability. There are three primary layers to be managed, although in some circumstances there may be more, and in other circumstances there may be less.

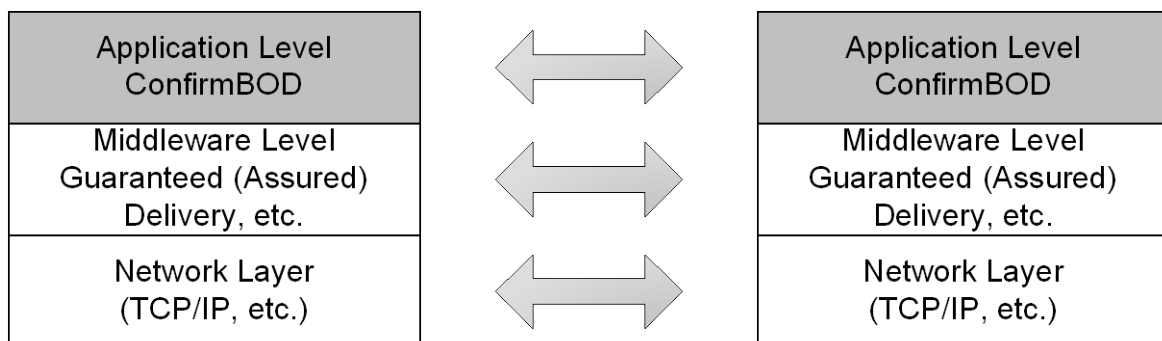


Figure 3 – The intended use of the ConfirmBOD

The ConfirmBOD ensures that the business applications or components have a standard communication mechanism in the case where the receiving application (of the original message) needs to communicate a response message back to the originating business application.

An instance of the ConfirmBOD communicates the processing results of another BOD Instance. BOD message processing refers to all of the system processes that execute as part of processing the message; this includes pre-processing such as message parsing and validation, in addition to message execution (of the operation it represents).

In summary, BODs are used to represent and communicate business actions and events across systems as part of realizing the business process distributed across the application landscape. The usage of the ConfirmBOD is subsequently reserved to support communication of the processing results of these messages.

The noun-independent or generic definition of the ConfirmBOD provides systems (i.e., business components) a mechanism for communicating message processing results (i.e., warning and error messages) in their own language. The flexibility of the ConfirmBOD schema offers adopters the opportunity to decide what should or should not be required in a ConfirmBOD instance.

Web Service Example

Figure 4, provides a graphical representation of how the BOD exchange fits within the basic Web Services stack. SOAP and HTTP address the Middleware Level and the BOD Exchange communicates at the Application Level.

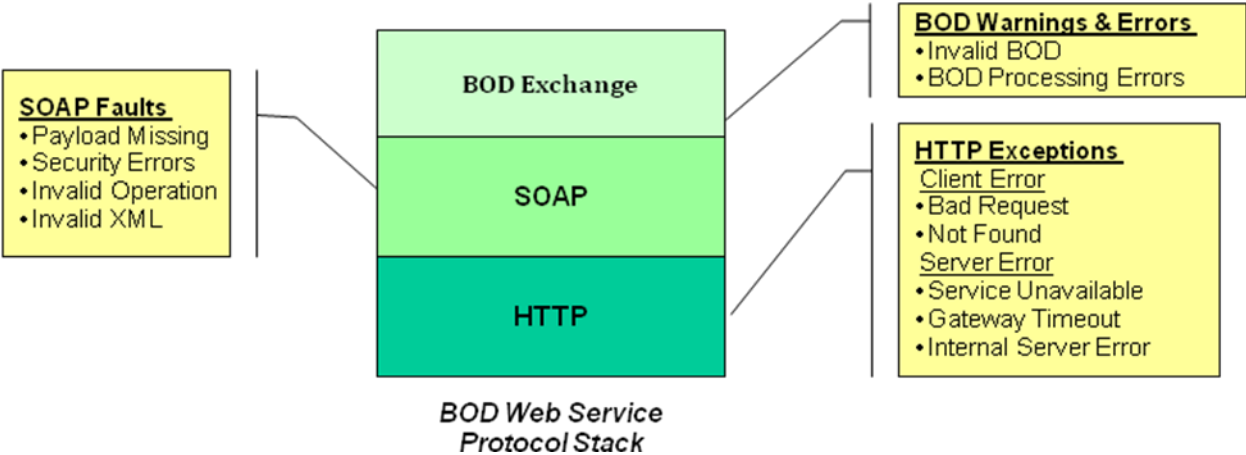


Figure 4: Errors across the Message Protocol Stack for Web Services

BODs, Verbs Paired with the ConfirmBOD

OAGIS defines a finite set of verbs with specific semantics. These verbs participate in common verb usage patterns that prescribe the recommended verb interaction (see Table 1) between communicating systems. Notice the presence of two types of response messages: noun-specific and noun-independent. The noun-specific response uses the same noun as that of initiating message. This facilitates noun-specific information to be returned in a response message. Conversely, the noun-independent response, ConfirmBOD, does not use the same noun as that of the initiating message; rather it is a general response that may be used across all of noun-specific messages. The ConfirmBOD facilitates the return of message processing success, warning, and failure information.

Table 1 provides a list of Verb used with given BODs in which the ConfirmBOD may be provided as a response.

OAGIS 9.X MESSAGES	RESPONSE MESSAGES
--------------------	-------------------

	NOUN-SPECIFIC	NOUN-INDEPENDENT
Post[noun] (synonym for Process in financial scenarios)	Acknowledge[noun]	ConfirmBOD
Process[noun]	Acknowledge[noun]	ConfirmBOD
Cancel[noun]	None	ConfirmBOD
Load[noun] (synonym for Sync in financial scenarios)	None	ConfirmBOD
Notify[noun]	None	ConfirmBOD
Sync[noun]	None	ConfirmBOD
Change[noun]	Respond[noun]	ConfirmBOD
Update[noun]	Respond[noun]	ConfirmBOD
Get[noun]	Show[noun]	ConfirmBOD
Acknowledge[noun]	None	ConfirmBOD
ConfirmBOD	None	None
Respond[noun]	None	ConfirmBOD
Show[noun]	None	ConfirmBOD

Table 1: OAGIS 9.x Verb Usage Pattern

[Note] The ConfirmBOD is generally available as a response to a BOD using any of the verbs (with the exception of a Confirm; in other words, one would not issue a ConfirmBOD as a result of receiving a ConfirmBOD).

How to use the ConfirmBOD

A ConfirmBOD may be requested and if so it must then be produced. The following section describes:

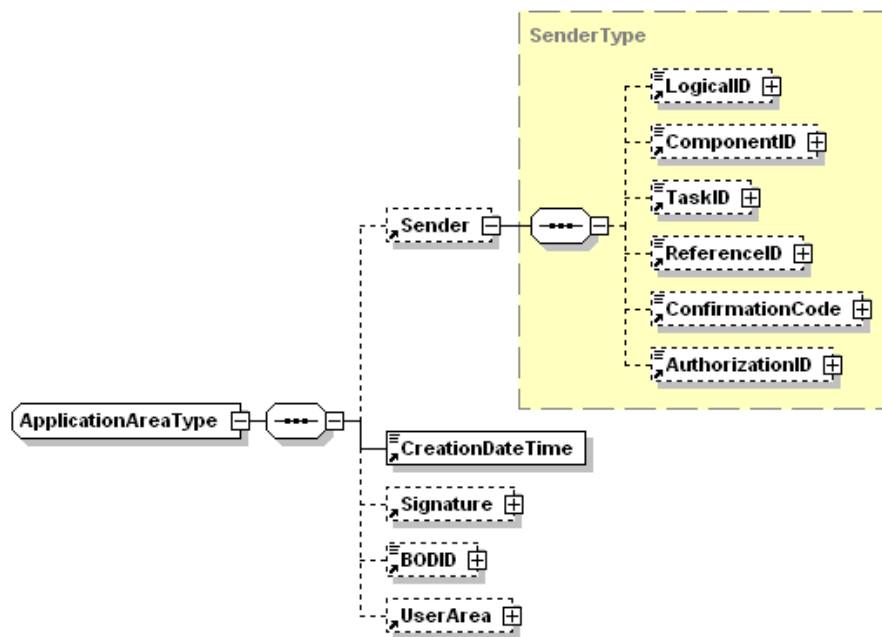
- Requesting a ConfirmBOD
- Producing a ConfirmBOD

Requesting a ConfirmBOD

A Sender may request a ConfirmBOD through the Sender element of any BOD's ApplicationArea, shown in Figure 5. The Sender element provides mechanism, the ConfirmationCode, by which to request a ConfirmBOD. This ConfirmationCode represents the sending system's results criterion upon which a ConfirmBOD should be returned.

[OAGi R 1]

The sending system for all other BODs other than the ConfirmBOD MUST provide a ConfirmationCode that specifies when a ConfirmBOD is to be returned.



Generated by XmlSpy

www.altova.com

Figure 5 – Application Area Type and the Sender element expanded.

[Note] A ConfirmBOD request, in addition to specifying when a ConfirmBOD instance is to be returned, must also used to specify that no ConfirmBOD instance be returned.

The ConfirmationCode

The ConfirmationCode specifies the *results criterion* of the message instance processing (i.e., error) determined by the receiving system.

There are several points *during* the processing of a BOD instance, for example: Receipt, Validation, and Completion. The term *processing*, here, refers to the receiving system's pre-processing of the BOD instance

from receipt through execution of the BOD instance. In addition, the term refers to processing at the BOD Exchange Layer and *not* any processing that may be occurring in lower layers of the protocol stack (see Figure 1). For example, although a transport layer protocol may assure ConfirmBOD delivery (i.e., message instance arrival to a particular queue), this does not guarantee receipt of the message instance to the processing system. Only the receiving system can confirm the proper delivery and receipt of the message instance.

By definition, the ConfirmationCode is associated with the *Completion* of BOD processing.

[OAGi R 2]

The possible values of the ConfirmationCode element are:

“Always” - The ConfirmBOD should always be sent.

“OnError” - The ConfirmBOD should only be sent on error.

“Never” - The ConfirmBOD should never be sent.

Producing a ConfirmBOD

A ConfirmBOD response is sent by the receiving system as instructed by the ConfirmationCode value in the BOD instance as specified by the sending system.

A ConfirmBOD response always corresponds to the completion of processing of the message instance. At most a single ConfirmBOD instance may be sent upon the completion of a specific processing step during message processing. In other words, multiple ConfirmBODs should not be sent for processing “parts of” a message instance (i.e., the first half of the message instance followed by the second half of the message instance).

[OAGi R 3]

The receiving system **MUST** produce *only one* ConfirmBOD instance upon processing completion if requested.

[Note] The specific cases for when a receiving system must or must not produce a ConfirmBOD instance are provided below: If the results criterion is “Always” then the receiving system **MUST** produce exactly one ConfirmBOD instance upon processing completion. If the results criterion is “OnError” and the processing resulted in success, then the receiving system **MUST NOT** produce a ConfirmBOD instance. If the results criterion is “OnError” and the processing resulted in errors, then the receiving system **MUST** produce a ConfirmBOD instance.

329

330

BOD and Noun Messages in ConfirmBOD

331

Messages in the ConfirmBOD may be conveyed at two-levels: the BOD-level and the noun-level. A noun-level message is always within the context of a given BOD instance.

332

333

334

[OAGi R 4]

335

All BOD and Noun messages MUST pertain to the processing of the BOD Instance in which the ConfirmBOD request was communicated.

336

337

338

BOD Messages

339

At the BOD-level, the ConfirmBOD supports the following messages (see Figure 6):

340

341

- BODFailureMessage

342

- BODSuccessMessage

343

- PartialBODFailureMessage

344

BODFailureMessage

345

A BODFailureMessage, shown in Figure 6, is created when the processing of the BOD fails. The message provides a list of error and warning messages that explain the failure.

346

347

348

Error and warning messages are represented by the

349

ErrorProcessMessage and WarningProcessMessage

350

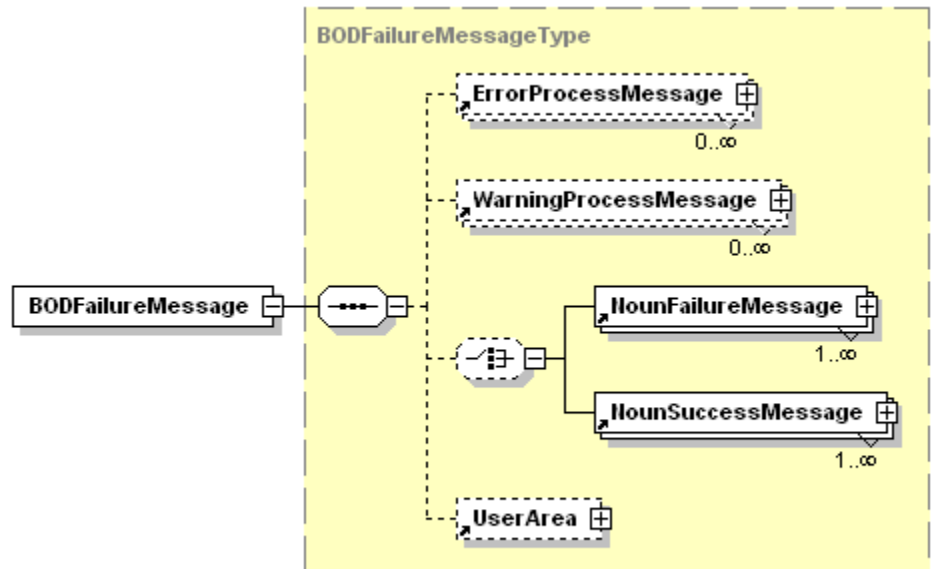
elements, respectively. The specific results from processing

351

each noun are reported in the NounFailureMessage and

352

NounSuccessMessage.



Generated by XmlSpy

www.altova.com

Figure 6 - BODFailureMessage Element of the ConfirmBOD

[Note] The choice compositor between the NounFailureMessage and NounSuccessMessage disallows the coexistence of both NounFailureMessage and NounSuccessMessage elements in a ConfirmBOD response.

[OAGi R 5]

BOD-level error or warning messages associated with a given BOD failure MUST be communicated in the ErrorProcessMessage and WarningProcessMessage elements, respectively within the BODFailureMessage element.

[OAGi R 6]

Noun-level failure messages MUST be communicated in the NounFailureMessage element of the BODFailureMessage element.

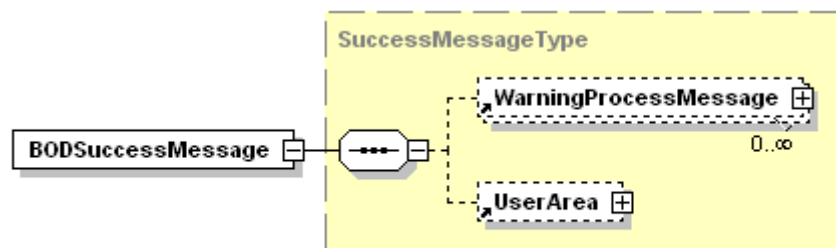
[OAGi R 7]

If Noun-level failures do not exist then Noun-level warning messages if they exist MUST be communicated in the NounSuccessMessage element of the BODFailureMessage element.

BODSuccessMessage

A BODSuccessMessage, shown in Figure 7, is created when the processing of the BOD was a success. Non-fatal

375 warning messages are also possible. A warning message is
376 represented by the WarningProcessMessage element.



377 Generated by XmlSpy www.altova.com

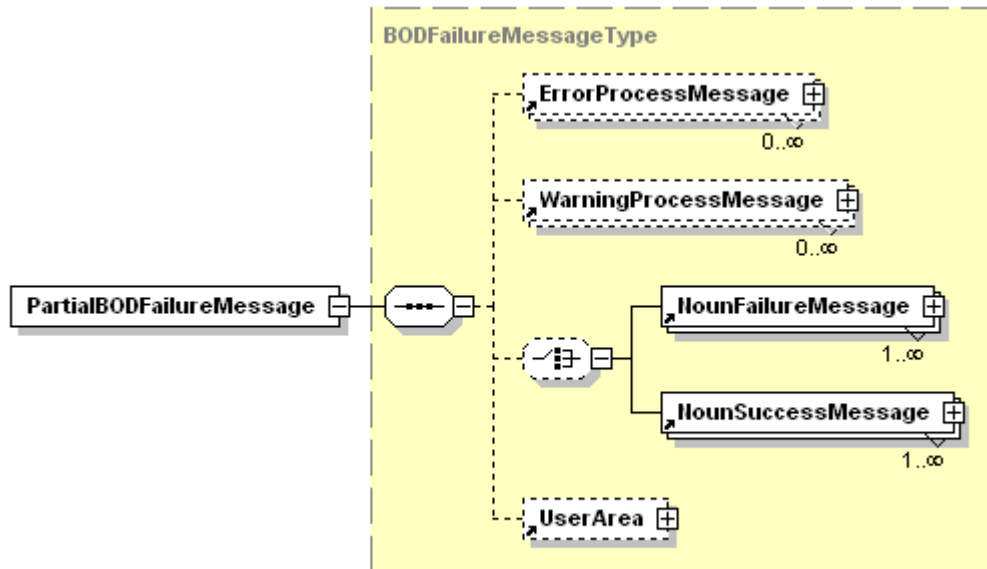
378 **Figure 7: BODSuccessMessage Element of the ConfirmBOD**

379 [OAGi R 8]
380 In the absence of BOD-level warning messages, the BODSuccessElement
381 MUST be communicated as an empty element.
382 [OAGi R 9]
383 BOD-level warning messages associated with the BOD success MUST be
384 communicated in the WarningProcessMessage element of the
385 BODSuccessMessage element.

386

387 **PartialBODFailureMessage**

388 A PartialBODFailureMessage, shown in Figure 8, is created
389 when the processing of at least one noun in the BOD has
390 failed. The message provides a list of error and warning
391 messages that explain the failure. Error and warning
392 messages are represented by the ErrorProcessMessage
393 and WarningProcessMessage elements, respectively. The
394 specific results from processing each noun are reported in
395 the NounFailureMessage and NounSuccessMessage.



Generated by XmlSpy

www.altova.com

Figure 8: PartialBODFailureMessage Element of the ConfirmBOD

[OAGi R 10]

BOD-level error or warning messages associated with a BOD instance partial failure MUST be communicated in the ErrorProcessMessage and WarningProcessMessage elements, respectively, in the PartialBODFailureMessage element.

[OAGi R 11]

Noun-level failure messages MUST be communicated in the NounFailureMessage element of the PartialBODFailureMessage element.

[OAGi R 12]

If Noun-level failures do not exist, then Noun-level warning messages if they exist MUST be communicated in the NounSuccessMessage element of the PartialBODFailureMessage.

Note: The choice compositor between the NounFailureMessage and NounSuccessMessage disallows the coexistence of both NounFailureMessage and NounSuccessMessage elements in a ConfirmBOD response.

Noun Messages

At the noun-level, the ConfirmBOD supports the following messages (see Figure 9):

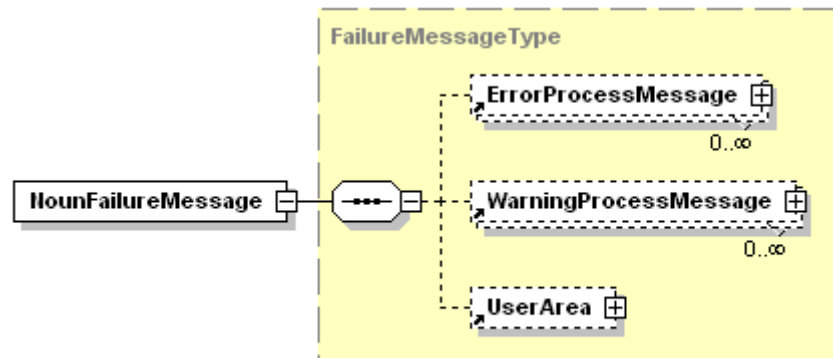
- NounFailureMessage

- NounSuccessMessage

Both of these messages are used to represent the results from processing of a noun.

NounFailureMessage

The NounFailureMessage is available for use in two of the BOD-level messages: BODFailureMessage and PartialBODFailureMessage. A NounFailureMessage, shown in Figure 6, is created when the processing of the noun fails. The message provides a list of error and warning messages that explain the failure. Error and warning messages are represented by the ErrorProcessMessage and WarningProcessMessage elements, respectively.



Generated by XmlSpy

www.altova.com

Figure 9: NounFailureMessage Element of the ConfirmBOD

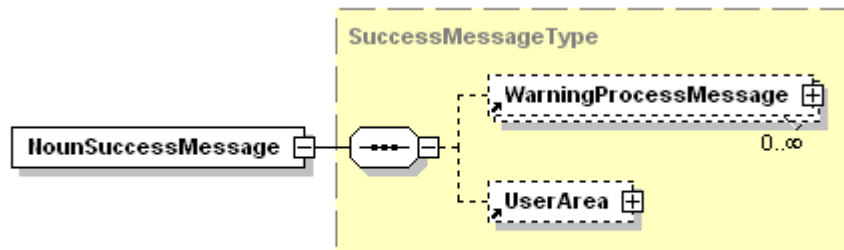
[OAGi R 13]

Noun-level error or warning messages associated with a given Noun failure MUST be communicated in the ErrorProcessMessage and WarningProcessMessage elements respectively of the NounFailureMessage element.

NounSuccessMessage

The NounSuccessMessage is available for use in all of the BOD-level messages: BODFailureMessage, BODSuccessMessage and PartialBODFailureMessage. A NounSuccessMessage, shown in Figure 10, is created when the processing of the noun was a success with warnings. Processing of a noun may result in non-fatal warning

messages. These warning messages are represented by WarningProcessMessage element.



Generated by XmlSpy

www.altova.com

Figure 10: NounSuccessMessage Element of the ConfirmBOD

[OAGi R 14]

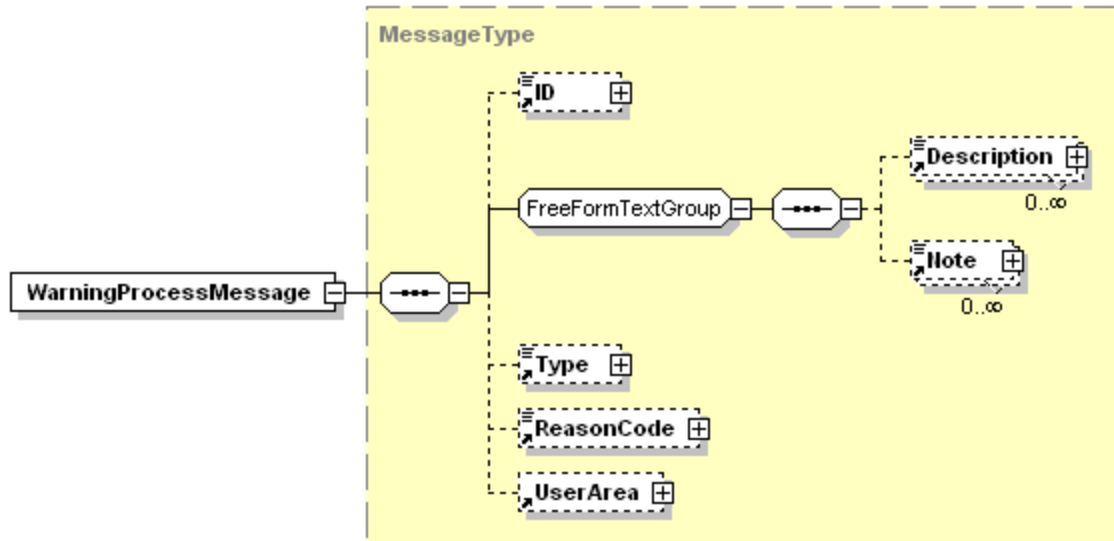
Noun-level warning message associated with the Noun success MUST be communicated in the WarningProcessMessage element of the NounSuccessMessage element.

Warning and Error Messages

Both the warning (WarningProcessMessage) and error (ErrorProcessMessage) have identical content models. Figure 11, below, shows the WarningProcessMessage.

The elements of the WarningProcessMessage and ErrorProcessMessage are defined as follows:

- ID – identifier of the warning or error message
- Description – description of the warning or error
- Type – type of warning or error
- ReasonCode – reason code for the warning or error



Generated by XmlSpy

www.altova.com

Figure 11: WarningProcessMessage Element of the ConfirmBOD

[OAGi R 15]

Values MUST be assigned to the following elements: ID and Description.

Identifying BOD Instances

The receiving system must relate the ConfirmBOD response that it produces to the BOD instance that it processed. This allows the sending system to relate the processing results, described in the ConfirmBOD, with the BOD instance that it sent. This is accomplished through the use of an identifier.

A BOD instance must be assigned an identifier; this is accomplished with the BODID element located in the ApplicationArea of any BOD. The ConfirmBOD, similar to the other response messages, provides an OriginalApplicationArea to communicate the BODID of the BOD instance from which the ConfirmBOD request was made. In the ConfirmBOD, the OriginalApplicationArea is located in the Confirm element.

[OAGi R 16]

The BODID of the OriginalApplicationArea element MUST be used to identify the BOD instance for which the ConfirmBOD was made. The BODID is provided in every BOD instance.

Identifying Noun Instances

In the case of multiple noun instances, a noun instance may be identified in two ways:

- DocumentID – is a common identifier available in nouns that are considered documents (e.g., PurchaseOrder); note that this identifier is not available across all nouns.
- ID – is a common identifier available in nouns that are not considered documents.
- Noun Instance Sequence – is a number that refers to the position of a noun instance in a BOD instance. For example, if a BOD instance were to contain 2 noun instances, the first occurrence is considered to be in the first position and may be identifier with a sequence number.

If more than one noun instance is sent in a BOD message instance, then the DocumentID and/or DocumentSequence must be used to differentiate the noun instances.

[OAGi R 17]

The DocumentID, ID, or Noun Instance sequence MUST be used to identify the Noun instance.

Note: These techniques for noun instance identification are intended to be used in the source data identification of error and warning messages.

The noun instance identifier may be communicated as part of the description or note of the warning or error process message.

ConfirmBOD Best Practices (non-Normative)

This section is intended to give the OAGIS® user some advice on some of the best ways to take advantage of the ConfirmBOD in your OAGIS® deployment.

Data Integrity

The ConfirmBOD was originally designed by the OAGi members by several Certified Public Accountants (CPAs). It is a generally accepted best practice to include at least one CPA on a business application design team. This practice usually increases the quality of the software design because the CPA trained person includes strong disciplines for building controls, audit functions, and cross checks into the business software.

The CPA's on the original OAGIS® Architecture team gave us the ConfirmBOD for this purpose, and it is recommended that it be used whenever data integrity is of importance, which is to say, all of the time.

523 It is strongly recommended that any integration scenario that requires a high order of
524 application data integrity use the Confirm BOD when not applying another type of
525 response.

526 **Multiple Noun option for OAGIS® BODs.**

527 The OAGIS® BOD architecture provides for sending multiple instances of a Noun in a
528 single BOD structure. An example of this might be sending a SyncPurchaseOrder with
529 100 occurrences of the PurchaseOrder Noun in the BOD instance.

530 This is a convenient way to send multiple occurrences of data without having to send
531 multiple BODs. There is one drawback to using this feature of the BOD architecture,
532 however. If one is using the ConfirmBOD and wants a response for a specific instance of
533 one of the Nouns communicated in the multiple instances ProcessPurchaseOrder BOD,
534 the ConfirmBOD can not discern which instance of the Noun was the exception.

535 This is because the exception processing is done at the BOD level, and this is the level
536 that the globally unique identifier (GUID) provided for in the BOD architecture resides.

537 Therefore it is recommended that if you are planning to use the multiple instance of Nouns
538 option, please keep this limitation in mind and consider using single instance Nouns in
539 your BODS to maintain full integrity with your ConfirmBOD processing.

540 **Publish and Subscribe Error Handling**

541 OAGIS is often used within a publish and subscribe communication model. In this
542 situation, a generic error management facility has proven useful. There are two
543 perspectives: 1. Centralized management of errors as they occur. 2. Centralized
544 monitoring for error conditions.

545

546 **Centralized Error Management**

547 In this scenario, a generic Error Log service captures all error conditions by
548 subscribing to ConfirmBOD error messages published on the so-called
549 Enterprise Service Bus (ESB). This does not preclude or exclude error
550 handling by the applications themselves. However, it does provide a
551 convenient means to manage errors to discover systemic problems. For
552 example, if an invalid UOM value is introduced by a single application, its
553 effect on other applications can be detected and the root cause easily
554 discovered. Contrast this with individual application level error handling and
555 root cause analysis, it can easily be seen that this approach requires less
556 manual intervention.

557

558 **Centralized Monitoring**

559 Although this approach does not explicitly involve the ConfirmBOD it does
560 augment both Application level and Centralized Error Management efforts.
561 A generic Monitoring Service known as the “BOD Validator” subscribes to
562 all BODs published on the ESB. Each message is validated for syntactic,
563 structural and content integrity. Syntactic integrity checks for proper XML
564 syntax. Structural integrity checks validate the message with the DTDs or
565 Schemas. Finally, content integrity validates data content against a known
566 dictionary (valid values) and interdependency (constraints) among
567 elements not expressed in the DTDs or Schemas. The latter leverages
568 Schematron and a file-based XML data dictionary. Error conditions are
569 centrally logged within the ESB.

570 The log is available to application owners and developers as an aid to
571 debugging the error condition. In some situations, the error message is
572 emailed to the originating application owner to help determine the source of
573 the error. The log also serves as an independent source of information.
574

575 Data Synchronization Integrity

576 The Sync verb is often used in application-to-application integration scenarios to
577 synchronize data from an owning application to other applications interested in the subject
578 data (e.g. Item). In the situation where multiple applications are receiving the
579 synchronized data, the ConfirmBOD can be used to provide transactional integrity among
580 all of the interested applications. When each receiving application responds with a positive
581 ConfirmBOD then the originating application successfully disseminated the data and
582 integrity is maintained. When any of the receiving applications responds with a negative
583 ConfirmBOD then alternate actions must take place to ensure integrity. This can be
584 automated by the originating application rolling back the other successful messages or by
585 raising an exception through some other mechanism outside of the OAG message
586 context.

587

Appendix A – Glossary

Application – Business software system

Message – Definition or schema of the information from which message instances are instantiated.

BOD – (Business Object Document) is a message that assembled from OAGi's Integration Specification (OAGIS)

ConfirmBOD – Definition or schema of the information used to describe message processing results.

Message instance – An instance of message that complies with the definition or schema of the message.

BOD instance – Is a *message instance* of a BOD.

ConfirmBOD instance – An instance of the ConfirmBOD.

Confirm BOD request – A request for *zero* or more ConfirmBOD instances communicated in a BOD instance (that is not a ConfirmBOD instance).

ConfirmBOD response – A ConfirmBOD instance sent in response to a ConfirmBOD request; note that multiple responses are possible from a single request.

Sending System – System that produces and sends a BOD instance (other than a ConfirmBOD instance) that communicates a ConfirmBOD request

Receiving System – System that receives and consumes a BOD instance from a sending system and produces a ConfirmBOD response.

Appendix B: Sample System Flows

Note: For the elements referenced in the examples of this appendix, <http://www.openapplications.org/oagis/9> is assumed to be the default namespace of the BOD instance.

B.1 Scenario 1

A sending system requests ConfirmBOD instances from the receiving system upon BOD instance processing, whether successful or not.

- Sending system sets the confirmation codes in the ApplicationArea element of the BOD instance:
 - `/ApplicationArea/Sender/ConfirmationCode="Always"`
`@listName="ResponseCodeContentType" @listAgencyName="OAGi"`
- Sending system sends the BOD instance
- Receiving system receives the BOD instance
- Receiving system finishes processing of the BOD instance successfully.
- Receiving system produces a ConfirmBOD instance upon processing completion of the BOD instance
 - Receiving system sets the OriginalApplicationArea in the Confirm element; this allows the receiver of the ConfirmBOD to relate the ConfirmBOD to the BOD instance that requested the ConfirmBOD.
 - `/ConfirmBOD/DataArea/BOD/OriginalApplicationArea`
 - Receiving system sets the BOD and Noun specific warning and error messages as needed
- Receiving system sends the ConfirmBOD instance

B.2 Scenario 2

A sending system requests ConfirmBODs from the receiving system upon BOD instance processing, on error only.

- Sending system sets the confirmation codes in the ApplicationArea element of the BOD instance:
 - `/ApplicationArea/Sender/ConfirmationCode="OnError"`
`@listName="ResponseCodeContentType" @listAgencyName="OAGi"`

- 641
 - Sending system sends the BOD instance
- 642
 - Receiving system receives the BOD instance
- 643
 - Receiving system finishes processing of the BOD instance and detects warnings
- 644
 - and errors.
- 645
 - Receiving system produces a ConfirmBOD instance upon completion of the BOD
- 646
 - instance
- 647
 - Receiving system sets the original application area in the Confirm element;
- 648
 - this allows the receiver of the ConfirmBOD to relate the ConfirmBOD to the
- 649
 - BOD instance that requested the ConfirmBOD.
- 650
 - [/ConfirmBOD/DataArea/BOD/OriginalApplicationArea](#)
- 651
 - Receiving system sets the BOD and Noun specific warning and error
- 652
 - messages as needed
- 653
 - Receiving system sends a ConfirmBOD instance
- 654

Appendix C: Success and Error Scenarios

This section provides samples of the BOD and Noun-level messages based on a BOD instance processing results. They are intended to help illustrate what Message elements should be sent in a ConfirmBOD given a certain processing result (success, failure, etc.).

[Note] For the elements referenced in the examples of this appendix, <http://www.openapplications.org/oagis/9> is assumed to be the default namespace of the BOD instance.

C.1 Scenario 1

Entire BOD instance was successfully processed.

- Existence of the BODSuccessMessage tag indicates that the entire BOD was received and processed successfully.
 - [/ConfirmBOD/DataArea/BOD/BODSuccessMessage](#)

C.2 Scenario 2

Entire BOD instance failed processing.

- Existence of the BODFailureMessage tag indicates that the entire BOD was received and failed to process.
 - [/ConfirmBOD/DataArea/BOD/BODFailureMessage](#)
 - Communicate Error Message(s):
 - [/ConfirmBOD/DataArea/BOD/BODFailureMessage](#)
 - [/ErrorProcessMessage](#)

C.3 Scenario 3

Entire BOD instance was successfully processed with warnings.

- Existence of the BODSuccessMessage tag indicates that the entire BOD was received and processed successfully.
 - /ConfirmBOD/DataArea/BOD/BODSuccessMessage
 - Communicate Warning Message(s):
 - /ConfirmBOD/DataArea/BOD/BODSuccessMessage
 - /WarningProcessMessage

C.4 Scenario 4

One or more noun instances in the BOD instance failed to process.

- Existence of the PartialBODFailureMessage tag indicates that a portion of the BOD was processed successfully
 - /ConfirmBOD/DataArea/BOD/PartialBODFailureMessage
 - Communicate Error Messages at the Noun-level for the noun instance(s) that failed to process:
 - /ConfirmBOD/DataArea/BOD/PartialBODFailureMessage
 - /NounFailureMessage/ErrorProcessMessage
 - Communicate Warning Messages at the Noun level for the noun instance(s) the processed successfully with warnings.
 - /ConfirmBOD/DataArea/BOD/PartialBODFailureMessage
/NounSuccessMessage/WarningProcessMessage