

MANAGED ACCOUNTS SERVICE  
MAS SYSTEM OVERVIEW



---

# Table of Contents

**DTCC MAS System Introduction ..... 3**

**Messaging Business Process..... 4**

**DTCC MAS Messaging Process..... 5**

**Combined Business and Messaging Process..... 6**

**Using The DTCC MAS System – Functional Process ..... 7**

**Using The DTCC MAS System – Step-By-Step Functional Process ..... 7**

**Step 1: In Brief ..... 7**

**The Sponsor Side – Step 1 ..... 7**

**The IM Side – Step 1 ..... 8**

**Step 2: In Brief ..... 8**

**The IM Side – Step 2 ..... 8**

**The Sponsor Side – Step 2 ..... 8**

**Step 3: In Brief ..... 9**

**The IM Side – Step 3 ..... 9**

**The Sponsor Side – Step 3 ..... 9**

**Business Messages ..... 10**

**Acknowledgement Messages ..... 11**

**Accept/Reject Messages..... 12**

**SOAP and DTCC MAS Messages ..... 12**

**Introduction to SOAP ..... 12**

**SOAP Message Components ..... 13**

**Illustration of SOAP Message Structure ..... 14**

**Glossary of Message Elements ..... 15**

---

## DTCC MAS System Introduction

The DTCC Managed Accounts Service (MAS) standardizes and automates the way in which managed accounts are set up and maintained on the systems of industry Sponsors and Investment Managers (IMs). The Managed Accounts Service is an offering of DTCC Solutions LLC, a subsidiary of The Depository Trust & Clearing Corporation, and will be part of DTCC's line of business-to-business XML services within the Wealth Management Services division. DTCC's Managed Accounts Service enables sponsors and investment managers to communicate business notifications via a centralized, standardized platform.

DTCC MAS will function as a messaging router and repository, acting as a central messaging gateway between Sponsors and IMs. Managed Accounts industry participants can send messages to DTCC MAS, request basic information regarding the messages contained in the system, and collect messages from the system. DTCC is not responsible for the content of these messages, but acts as a central location for collecting and routing these messages.

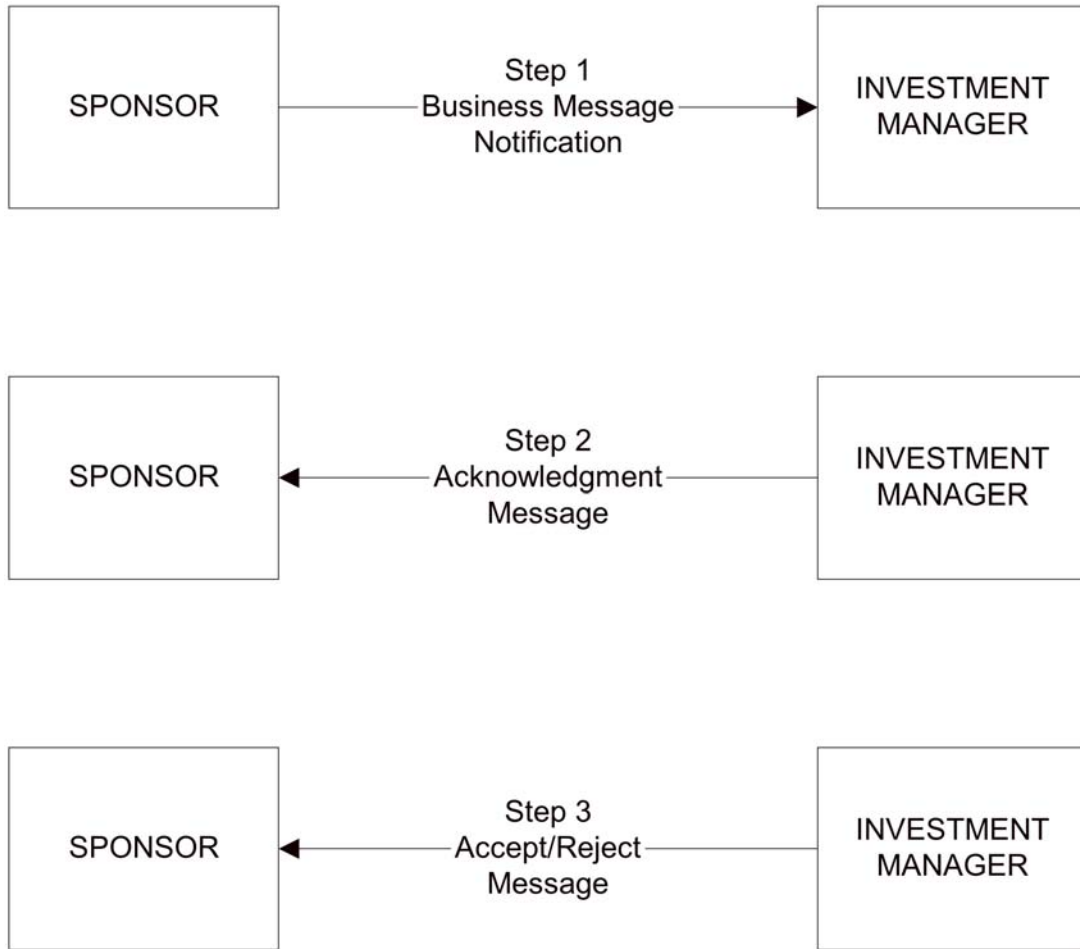
DTCC MAS will be a "client" facility that will allow each participant the flexibility to send and retrieve messages as frequently as desired. The Managed Accounts Service serves as a central link between trading parties, and automates the routing of account messages to and from the appropriate parties. The advantages of this approach are:

- As DTCC MAS clients, Sponsors and IMs are spared additional costs that are associated with operating and maintaining their own web servers.
- The message protocol allows participants to retrieve messages interactively at any frequency. It also supports dynamic frequency updates by giving feedback on how many messages are waiting in the queue.
- The message protocol provides secure routing to the intended party for business messages.
- The message protocol includes inquiry features to locate and retrieve a particular business message from the DTCC MAS system.

---

## Messaging Business Process

The messaging flow between participants is the following:



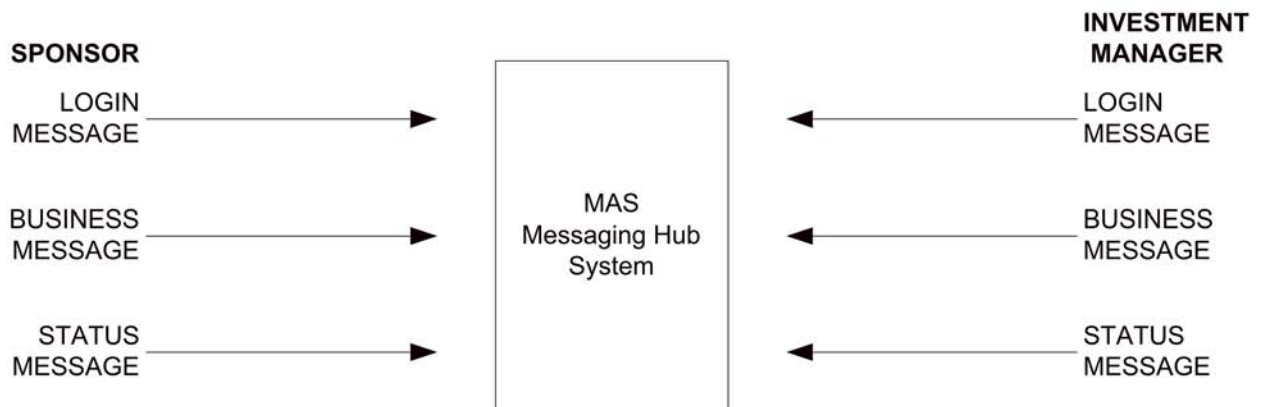
**Figure 1: DTCC MAS Messaging Process, Business View**

This view refers only to the two businesses on either side of a message. The sequence of events for the business flow is:

First	The Sponsor sends the IM a business message.
Second	The IM sends back an acknowledgement message, which simply confirms receipt of the Sponsor's business message.
Last	The IM sends an acceptance or rejection of the business message sent by the Sponsor.

## DTCC MAS Messaging Process

The relationship between the system and the users is illustrated by the following:



**Figure 2: DTCC MAS Messaging Process, System View**

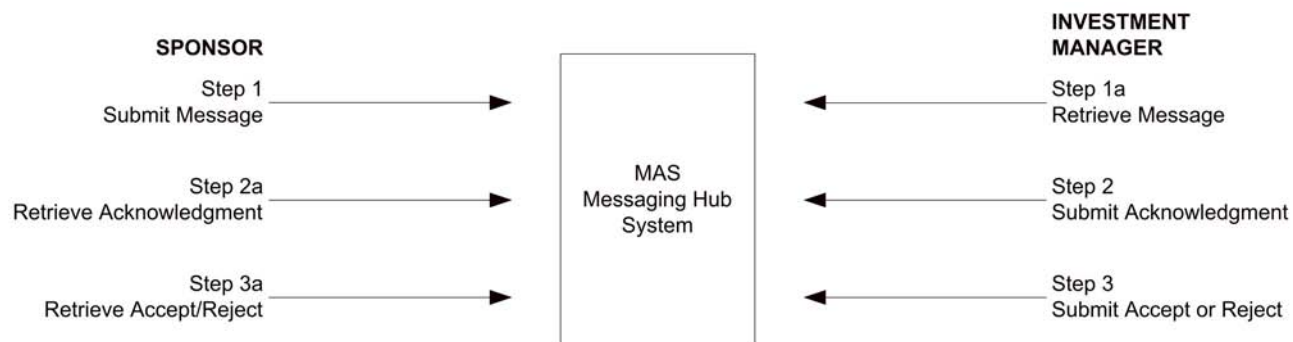
This diagram illustrates how the DTCC MAS system functions as a server repository for its clients.

Multiple processes and functions may occur simultaneously, and Sponsors and IMs may send and retrieve messages at any time. The hub uses a “pull” methodology, where messages reside in the repository until clients retrieve them. These messages include:

Login Message	Allows users to establish a session with the DTCC MAS System.
Business Message	Allows users to submit data into the DTCC MAS System.
Status Message	Allows users to retrieve data from the DTCC MAS System.

## Combined Business and Messaging Process

The procedure of exchanging business messages combines seamlessly with the DTCC MAS system's messaging process. Putting the two protocols together, the process looks like this:



**Figure 3: Business Messaging Process via DTCC MAS System**

The message flow through the DTCC MAS system is therefore broken down like this:

Step 1	Sponsor submits business message to DTCC MAS
Step 1a	IM sends a status message to MAS to retrieve business messages addressed to him in the repository.
Step 2	IM submits acknowledgement (a business message) to DTCC MAS
Step 2a	Sponsor sends a status message to MAS to retrieve the business (acknowledgement) message from DTCC MAS
Step 3	IM submits a business message to DTCC MAS to notify Sponsor of the accept/reject of the Sponsor's business message
Step 3a	Sponsor sends a status message to DTCC MAS to retrieve the business message with the accept/reject message from the IM

---

## Using the DTCC MAS System – Functional Process

The DTCC MAS system has a few general sequences to follow. The following table contains the basic functional process of messages through the system:

A	Users send a login message to log on to the MAS system.
B	Users send a business message to submit data into the system.
C	Users send a status message to retrieve data.
D	For every message sent to the system, DTCC MAS replies with a response message (request/response protocol).
E	DTCC MAS receives business messages and logs the information into a database. The database or queue is sorted according to the destination.
F	A user's status message receives a response message from DTCC MAS. DTCC MAS attaches the pertinent data stored in the database onto the response message for that particular user.

The actual exchange of messages can be seen in three basic steps with different actions for each party. The following pages discuss these steps in detail.

## Using the DTCC MAS System – Step-By-Step Functional Process

### Step 1: In Brief

In this step, the Sponsor sends a message regarding a customer to the DTCC MAS repository. The IM must then retrieve the message from the repository.

### The Sponsor Side – Step 1

The Sponsor sends a message on behalf of a customer to the DTCC MAS system. This message contains instructions and/or a request, as well as information that identifies where the message comes from, where the message is going, and when the message was sent.

The DTCC MAS system receives the message in step 1. At this time, the DTCC MAS system performs a validation process on that message. This process verifies that the above information (the source of the message, the destination of the message, and the time stamp) is recognized as valid by the DTCC MAS system. Note that the content itself is not checked for errors – only the validity of source and destination information about the message is checked.

- If the information is not accepted by the DTCC MAS system, the system generates an exception

- 
- message that is returned to the Sponsor.
  - If the information is validated by the system, then the message is placed in the DTCC MAS repository. It is not sent to the IM, but instead remains in the repository until the IM retrieves it.

### **The IM Side – Step 1**

To retrieve any messages addressed to it, the IM sends a status request to the DTCC MAS repository. The IM may send status requests frequently throughout the day in order to retrieve the most current information available. A request goes through the same sort of validation process as the Sponsor's message.

If the information in the status request message does not pass validation with the system, then a notification is sent back to the IM stating that its request message was invalid.

- If the message is validated by the system, then any messages addressed to the IM are pulled from the repository and delivered to the IM.

### **Step 2: In Brief**

In Step 1, the IM received a message from the Sponsor regarding a customer. In Step 2, the IM sends an acknowledgement to the repository confirming that it has received that message. The Sponsor must then retrieve the acknowledgement from the repository.

### **The IM Side – Step 2**

After the IM has received the message, the IM posts an acknowledgement to the DTCC MAS system. This message allows the IM to proactively verify that it has received the Sponsor's original message. It is only a statement that confirms these things:

- The IM sent a status request to the DTCC MAS repository in order to retrieve messages.
- The original message from the Sponsor, which was stored in the repository, was given to the IM in response to a status request.
- The IM received the message.

The acknowledgement message is posted to the DTCC MAS system by the IM. The system performs an acknowledgement validation process to determine if the message satisfies edit criteria. Note that the content itself is not checked for errors – only the information about the message is checked.

- If the message does not satisfy edit criteria, then the DTCC MAS system generates an exception message and sends it to the IM. This exception message states that the IM's original message was invalid.
- If the message satisfies edit criteria, then the IM's acknowledgement is sent on to the repository. The message is stored in the repository until the Sponsor sends another status request that asks for any unretrieved messages.

### **The Sponsor Side – Step 2**

To retrieve any messages and acknowledgements addressed to it, the Sponsor will send a status request to the DTCC MAS repository. The Sponsor may send status requests frequently throughout the day in order to retrieve the most current information available. A request will go through the same sort of validation process as an acknowledgement – only the information about the message is checked for validity.

- 
- If the status request information is not accepted by the system, then a notification is sent back to the Sponsor stating that its request message was invalid.
  - If the status request's information is validated by the system, then any messages addressed to the Sponsor will be pulled from the repository and delivered to the Sponsor.

### **Step 3: In Brief**

In Step 2, the IM sent an acknowledgement to the Sponsor that the original message was received. In Step 3, the IM sends either an accept or reject message to the Sponsor. The Sponsor must then retrieve the message from the repository and act upon the information contained in the message.

### **The IM Side – Step 3**

Once an acknowledgement to the original message is posted, the IM responds to the Sponsor's original message instructions. This message contains one of two possible responses to the request made on behalf of a customer:

- The IM accepts the request made by the Sponsor
- The IM rejects the request made by the Sponsor.

The accept/reject message is posted to the DTCC MAS system by the IM. The system performs the same validation process to determine if the message satisfies edit criteria. Note that the content itself is not checked for errors – only the information about the message is checked.

- If the message does not satisfy edit criteria, then the DTCC MAS system generates an exception message and sends it to the IM. This exception message states that the IM's original message was invalid.
- If the message satisfies edit criteria, then the IM's accept/reject message is sent on to the repository. The message is stored in the repository until the Sponsor sends another status request that asks for any unretrieved messages.

### **The Sponsor Side – Step 3**

To retrieve any accept/reject messages addressed to it, the Sponsor will send a status request to the DTCC MAS repository. The Sponsor sends status requests frequently throughout the day in order to retrieve the most current information available. A request will go through the same sort of validation process as the IM's accept/reject message – only the information about the message will be checked for validity.

- If the status request's information does not check out with the system, then a notification is sent back to the Sponsor stating that its request message was invalid.
- If the status request's information checks out with the system, then any accept/reject messages addressed to the Sponsor will be pulled from the repository and delivered to the Sponsor.

After the Sponsor receives the accept/reject message from the IM, the Sponsor then contacts their investor client and acts on that information accordingly. If the original message is accepted by the IM, then the Sponsor acts upon the customer's original instructions.

---

## Business Messages

Business messages include two basic message categories: Account Set-up messages and Account Maintenance messages.

The account set-up message process uses the Account Set-up schema but also mandates the new account funding notification and authorization to trade notification data tags within the New Account Set-up schema. DTCC MAS includes the following message types to support the new account set-up message process:

- New Account Set-up Notification
- Acknowledge Receipt of New Account Set-up Notification
- Accept or Reject New Account Set-up Notification

Account Maintenance messages include the following message types:

- Deposit/Contribution Funding Notification
- Acknowledge Receipt of Deposit/Contribution Notification
- Accept or Reject Deposit/Contribution Notification
- Withdrawal/Distribution Notification
- Acknowledge Receipt of Withdrawal/Distribution Notification
- Accept or Reject Withdrawal/Distribution Notification
- Change Profile Update Notification
- Acknowledge Receipt of Change Profile Update Notification
- Accept or Reject Change Profile Update Notification
- Account Termination Notification
- Acknowledge Receipt of Account Termination Notification
- Accept or Reject Account Termination Notification

All business messages follow the same general process flow:

1	Sponsor logs onto the system and establishes a session
2	Sponsor sends a business message (such as new account set-up notification) that contains the data elements necessary to establish an account with the IM.
3	DTCC MAS stores the data from the notification onto the database. Also, DTCC MAS returns a standard SOAP response message to the Sponsor.
4	IM logs onto the system and establishes a session.
5	IM sends a status message to retrieve data that belongs to that IM.
6	DTCC MAS reads the database for data that belongs to the IM, and attaches the data onto the response message to the IM.

---

## Acknowledgement Messages

Acknowledgement messages are sent by the IM to confirm that they have received a business message from the sponsor.

Types of acknowledgement messages include:

- New Account Set-up acknowledgement
- Deposit/Contribution Notification acknowledgement
- Withdrawal/Distribution Notification acknowledgement
- Change Profile Update Notification acknowledgement
- Termination Notification acknowledgement

All acknowledgement messages follow the same general process flow:

1	IM logs onto the system and establishes a session.
2	IM sends a business message (for example, a new account set-up acknowledgment), which acknowledges the receipt of the initial notification message.
3	DTCC MAS stores the data from the acknowledgment onto the database. Also, DTCC MAS returns a standard SOAP response message to the IM.
4	Sponsor logs onto the system and establishes a session.
5	Sponsor sends a status message to retrieve data that belongs to that Sponsor.
6	DTCC MAS reads the database for data that belongs to the Sponsor, and attaches the acknowledgement data onto the response message to the Sponsor.

---

## Accept / Reject Messages

Accept/reject messages are sent by the IM to inform the Sponsor whether the Sponsor's business message was accepted or rejected.

All accept/reject messages follow the same general process flow as acknowledgement messages:

1	IM logs onto the system and establishes a session.
2	IM sends a business message (for example, a new account set-up accept/reject) that accepts or rejects the initial notification message.
3	DTCC MAS stores the data from the accept/reject message onto the database. Also, DTCC MAS returns a response message to the IM.
4	Sponsor logs onto the system and establishes a session.
5	Sponsor sends a status message to retrieve data that belongs to that Sponsor.
6	DTCC MAS reads the database for data that belongs to the Sponsor, and attaches the accept/reject data onto the response message to the Sponsor.

## SOAP and DTCC MAS Messages

### Introduction to SOAP

The DTCC MAS application uses the Simple Object Access Protocol (SOAP) to route messages. SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML-based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and response.

SOAP has the following qualities:

- It defines a message format in XML, and can be implemented over the Hypertext Transfer Protocol HTTPS
- SOAP has the simplest set of specifications, the fastest development cycle, and an extensible request/response model
- SOAP is not tied to any operating system, programming language, or object model.

---

These features of SOAP have the following benefits:

- Users can customize their own interfaces to the DTCC MAS system (using their own object models), if desired
- IBM, SUN and Microsoft all support SOAP
- The messaging is kept simple, meaning that there is a smaller margin for error in the messaging process

## SOAP Message Components

A SOAP message is an XML document that contains three basic elements:

1. Envelope
2. Header
3. Body

The envelope is the top element of the XML document representing the message. A SOAP envelope is the container for all messages and defines the message protocol. The envelope is a simple schema composed of a single header tag and the body tag.

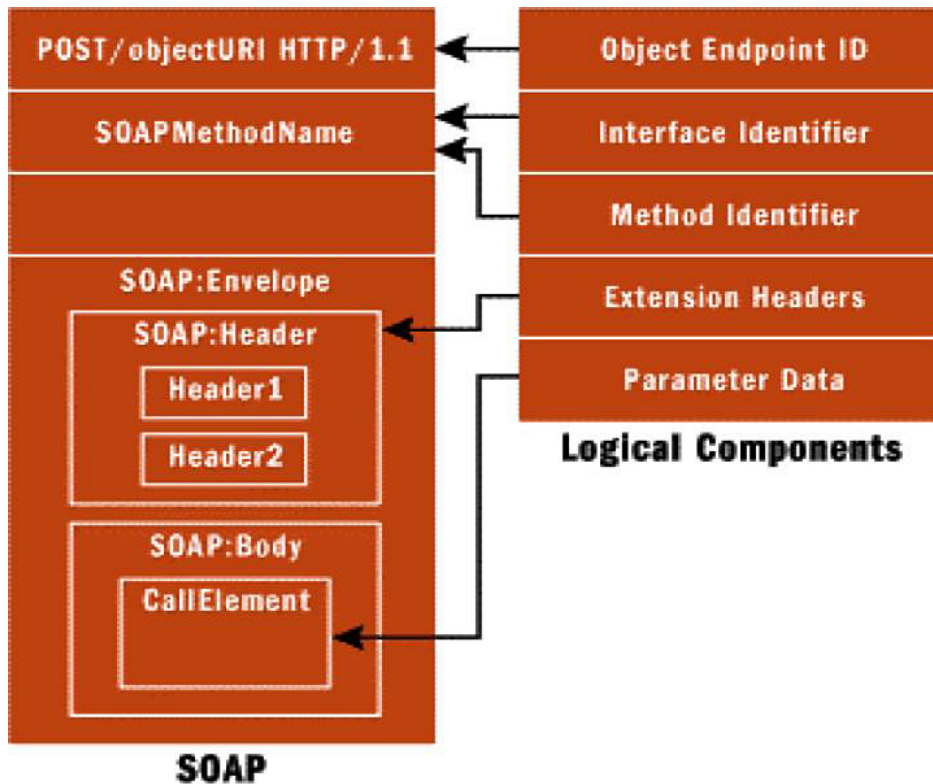
The header is a generic mechanism for adding features to a SOAP message in a decentralized manner without prior agreement between the communicating parties. The header handles basic information that can be understood by all parties using the DTCC MAS system, such as where to route the message, which parties have touched the message, and what type of message is enclosed in the envelope. SOAP defines a few attributes that can be used to indicate who should deal with a feature and whether it is optional or mandatory.

The header comprises the routing message (RM), which encapsulates a single Manifest tag and Delivery tag. These tags give the DTCC MAS system (as well as IMs and Sponsors) information about routing.

The body is a container for mandatory information intended for the ultimate recipient of the message. It encapsulates any of the SMA\_AS, SM\_AM, or SMA\_AD XML messages – in other words, the body is where you will put the actual content of your message.

DTCC is not responsible for the information contained in the body of a message. SOAP defines only one element for the body, which is the Fault element used for reporting errors.

## Illustration of SOAP Message Structure



**Figure 4: SOAP Message Structure**

The box on the left represents the protocol itself, comprising of general identifying information as well as the SOAP message. The SOAP message itself is sectioned into three elements.

The box on the right explains the different data elements (“logical components”) that are contained in the different areas of the protocol.

---

## Glossary of Message Elements

The XML language and its SOAP subset make use of a number of markers, called “tags”, that encapsulate the information contained in a message. Tags in XML, HTML, and other markup languages have the same basic structure:

- `<Tag>` indicates the beginning of a tag, and
- `</Tag>` indicates the end of a tag.

These tags can be seen as containers. Anything within these containers is an “element” that is described by those tags.

Using SOAP, anything contained within a pair of tags is marked as a unique piece of information. For example:

```
<Message>
  This is a sample message
</Message>
```

In this example, the text “This is a sample message” is marked up to actually be a message, since it exists within the message container.

As such, the entire SOAP envelope is a container:

```
<Envelope> indicates that this is the beginning of the container, and
</Envelope> indicates the end of the container.
```

The element “envelope” is described by these tags.

Anything between these tags is contained within the envelope. Therefore, if the envelope contains a message, it looks like this:

```
<Envelope>
  This is a message contained within the SOAP envelope.
</Envelope>
```

Tags can also be nested. This means that, in addition to containing the marked-up text, tags can also contain sub-tags that exist as separate pieces of information within a larger tag.

This can be applied to SOAP messages, which can simultaneously contain separate informational elements in a larger tag:

```
<Message>
  <MessageID>1234567</MessageID>
  This is a sample message
</Message>
```

---

This example means that, within a container called “Message,” there is the message itself as well as another piece of information, called a “MessageID.” This message ID is encapsulated within its own tags.

Therefore, the text “1234567” is a different piece of information from the text “This is a sample message,” although they exist within the same container. Both pieces of information are considered part of the larger message.