

# COSPAS-SARSAT Message Parser

## COSPAS-SARSAT Message Parser API

**Flight ATM Systems Ltd.**



Document Number  
**COSPAS-API**

Rev  
0.1

Page  
1/20

Filename: COSPAS\_Message\_Parser\_API d0.1.doc

Paper size: A4

All information contained in this document remains the sole and exclusive property of Flight ATM Systems Ltd. No part of it may be copied, or disclosed by the recipient to third persons, without the prior written consent of Flight ATM Systems Ltd.; nor shall it be used for any purpose other than in connection with an agreement or proposed agreement with Flight ATM Systems Ltd. Registered in the United Kingdom, registration number 562 5816;

## Table of Contents

1	Document Overview .....	4
1.1	Identification .....	4
1.2	Purpose.....	4
1.3	Intended Audience .....	4
1.4	Scope.....	4
2	Parser Overview.....	5
2.1	Supported Messages.....	5
2.2	Parser overview.....	6
2.3	Technical Features .....	6
2.4	Software Delivery .....	7
2.5	COSPAS-SARSAT Parser Data Flow .....	8
2.6	Error Handling .....	8
2.7	Configuration Files .....	9
2.7.1	Supported Messages File .....	10
2.7.2	Field Definition File.....	11
2.7.3	Sub-fields .....	12
2.7.4	Error Text File.....	12
2.8	W3C XML Output Document Structure.....	13
2.8.1	Accessing the Output Data.....	14
3	The API.....	15
3.1	Method Synopsis.....	15
3.1.1	Parser Instantiation.....	15
3.1.2	Parse Message .....	15
4	Appendix A – Acronyms .....	16
5	Appendix D – XSD For W3C Output Document .....	18

## List of Tables

Table 1 -	Supported COSPAS-SARSAT Messages .....	6
Table 2 -	Software Delivery Structure .....	7
Table 3 -	Descriptor Files Overview.....	9
Table 4 -	Supported Messages File Nodes and Attributes .....	10
Table 5 -	Field Definition File Nodes and Attributes .....	11
Table 6 -	Regular Expression File Nodes and Attributes.....	12
Table 7 -	Error Text File Node and Attribute Descriptions .....	12
Table 8 -	Output File Node and Attribute Descriptions.....	14
Table 9 -	Acronyms.....	17

---

## List of Figures

---

Figure 1 - COSPAS-SARSAT Message Parser Data Flow ..... 8  
 Figure 2 - Descriptor Files Relationship ..... 9  
 Figure 3 - W3C Output Document File Structure..... 13

---

## Referenced Documents

---

Referenc	Identification	Name
[1]	C/S A.002	COSPAS-SARSAT Mission Control Centres Standard Interface Description Issue 5 - Revision 5 October 2014

# 1 Document Overview

## 1.1 Identification

Product:	COSPAS-SARSAT Message Parser
Document Name:	COSPAS-SARSAT Message Parser API
Document Number:	COSPAS-API
Revision:	0.1
Revision Date:	Monday, 21 September 2015
Document Owner:	Peter Venton – Flight ATM Systems Ltd.
File Name:	COSPAS_Message_Parser_API d0.1.doc

## 1.2 Purpose

This document provides a description of the Flight ATM Systems Ltd. COSPAS-SARSAT Message Parser and its API.

## 1.3 Intended Audience

This document has been produced for programmers using the Flight ATM Systems Ltd. COSPAS-SARSAT Message Parser in order that they are able to integrate the parser into their own software products. It is expected that a reader have the pre-requisite knowledge in the Java programming language and an understanding of XML and XML Schema definitions.

## 1.4 Scope

This document describes the COSPAS-SARSAT Message Parser API and provides a broad overview of the parser functionality. The parser overview has been provided so as to provide background information on the concepts used to implement the parser.

This document is not a requirement document and does not attempt to specify topics in an atomic form.

## 2 Parser Overview

The parser can be configured to process any type of COSPAS-SARSAT message. By default, the parser is delivered with pre-configured support for all the messages defined in [1].

### 2.1 Supported Messages

The COSPAS-SARSAT Message Parser is able to process the COSPAS-SARSAT messages listed in the table shown below:

SIT	Description	Meaning
121	406 INTERFERER NOTIFICATION	This message is used for notification of 406 MHz interferer signals.
122	INCIDENT (NO DOPPLER)	A 406 MHz alert message with no Doppler positions. An encoded position may or may not be available.
123	406 POSITION CONFLICT (ENCODED ONLY)	A 406 MHz alert message with no Doppler positions for which the encoded position differs by more than the match criteria from all previous positions.
124	406 AMBIGUITY RESOLUTION (ENCODED ONLY)	A 406 MHz alert message with no Doppler positions that identifies the resolved position of a 406 MHz alert.
125	406 INCIDENT	A beacon alert message computed from 406 MHz incident data. The message contains Doppler positions.
126	406 POSITION CONFLICT	A beacon alert message computed from 406 MHz incident data. The message contains Doppler and/or encoded position(s), which may differ from previous position(s) by the match criteria.
127	406 AMBIGUITY RESOLUTION	A 406 MHz alert message with Doppler positions that identifies the resolved position of a 406 MHz alert. It may or may not contain an encoded position.
132	406 NOTIFICATION OF COUNTRY OF REGISTRATION (ENCODED ONLY)	This message is used between MCCs to notify the country of registration of a 406 MHz beacon (NOCR). This message contains only an encoded position.
133	406 NOTIFICATION OF COUNTRY OF REGISTRATION	This message is used between MCCs to notify the country of registration of a 406 MHz beacon (NOCR). This message contains Doppler positions. It may or may not contain an encoded position.
185	COSPAS-SARSAT ALERTS	This message is used for alert messages and as NOCR message between MCCs and SPOCs.
215	ORBIT VECTORS	Sarsat or Cospas spacecraft orbit position and time message.
216	ORBIT VECTORS	Sarsat or Cospas spacecraft orbit position and time message. Used in special conditions (e.g., after a satellite manoeuvre) when it is required that orbit vectors at the MCC and its associated LUTs be initialized.
415	SARP CALIBRATION	Time and frequency calibration for a SARP.
416	SARP TELEMETRY	SARP telemetry from a Sarsat spacecraft.
417	SARP-3 CALIBRATION	Time and frequency calibration for a SARP-3.
425	SARP OUT OF LIMIT	Warning message to indicate abnormal performance of the SARP.
435	SARP COMMAND	Command request for the SARP.
445	SARP COMMAND VERIFICATION	Verification of the execution (or non-execution) of a SARP command as requested by command message.

SIT	Description	Meaning
510	406 MHz SARR FREQUENCY CALIBRATION OFFSET	Offset between actual and 406 MHz SARR-provided beacon frequencies.
515	SARR TELEMETRY	SARR telemetry from a Sarsat spacecraft.
525	SARR OUT OF LIMIT	Warning message to indicate abnormal performance of the SARR.
535	SARR COMMAND	Command request for the SARR.
545	SARR COMMAND VERIFICATION	Verification of the execution (or non-execution) of a SARR command as requested by a SARR COMMAND message.
605	SYSTEM STATUS TO ALL MCCs	Narrative message transmitted to all MCCs to indicate changes in System status. System status messages include System element and System function failures, scheduled maintenance, integration or testing of new System elements, and the commissioning of new equipment or new capabilities of existing equipment.
915	FOR MCC INFORMATION TRANSMISSION TO A SINGLE MCC	Narrative message for MCC to MCC operator. This is a free format message, except when a specific format is defined (Note 1).
925	406 BEACON REGISTRATION INFORMATION	This message is used between MCCs to provide 406 MHz beacon registration information

**Table 1 - Supported COSPAS-SARSAT Messages**

**Note 1:** Free format applies only to the message text. The complete message must still be formatted as per the host communication networks procedures.

## 2.2 Parser overview

The message content, (the message fields) are specified in offline configuration data, which provides a mechanism to specify the supported messages and their content.

The field parser is implemented as a table driven finite state machine. The state machines are defined in configuration data providing high levels of flexibility to both modify and add/delete messages.

## 2.3 Technical Features

The COSPAS-SARSAT message parser requires a JRE 1.8.x to run. Other technical features include:

- Written in Java™;
- Platform independent;
- Output generated in an open standard W3C XML document;

## 2.4 Software Delivery

The software is delivered in zip files for windows and tar files for Unix based operating systems. The directory structure of the delivered software is as follows:

Directory	Description
lib/	All jar-files are located here
doc/	Location of the COSPAS-SARSAT Parser documentation (this file)
doc/cospas/flightatm/cospas/xml/descriptors	Root for COSPAS-SARSAT configuration files
doc/cospas/flightatm/cospas/xml/descriptors/ COSPAS_Output_Document.xsd	XML Schema describing the parser output
doc/api	Java doc for the parser interface
README.txt	Readme file
LICENSE.txt	Licensing agreement

**Table 2 - Software Delivery Structure**

## 2.5 COSPAS-SARSAT Parser Data Flow

The following figure illustrates the data flow through the parser and the association of the configuration data with the various software components.

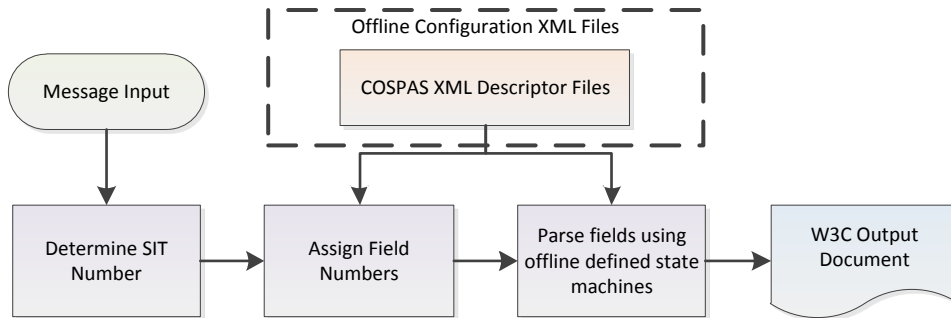


Figure 1 - COSPAS-SARSAT Message Parser Data Flow

## 2.6 Error Handling

Errors detected by the parser are included in the W3C XML output document; it is the caller’s responsibility to check if errors are present. How this can be checked is explained in section 2.8.1 titled ‘Accessing the Output Data’.



## 2.7 Configuration Files

There are four configuration files that define the supported COSPAS-SARSAT message numbers, (the SIT) and their content. These files are delivered with the parser software and can be found in the following directory:

`doc/cospas/flightatm/cospas/xml/descriptors`

These files can be used to configure any COSPAS-SARSAT message. By default the COSPAS-SARSAT Parser uses the class loader to read the COSPAS-SARSAT configuration files. The name space used is:

`flightatm.cospas.xml.descriptors`

One of the XML files in the 'descriptors' directory contains the text for all the errors reported by the parser, further information about these error messages can be found in section 2.7.4'.

The default COSPAS-SARSAT configuration files are packaged into the `cospas-configuration-X.jar`.

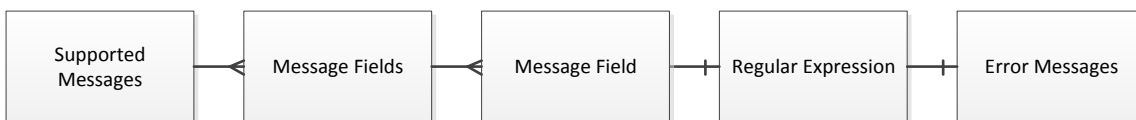
Setting the system property (Java -d) `flightatm.cospas.config.dir` tells the COSPAS-SARSAT Parser to load the COSPAS-SARSAT configuration files from the directory corresponding to the value of this property.

An overview of the COSPAS-SARSAT parser configuration files is provided in the following table, a more detailed description of each file is provided in the following sub-sections:

File Name	Description
<code>COSPAS_Supported_Messages.xml</code>	Contains a list of supported COSPAS-SARSAT messages along with the message content;
<code>COSPAS_Field_Definitions.xml</code>	Contains a list of the COSPAS-SARSAT message fields and the associated state machine for the field parsing;
<code>COSPAS_Regular_Expressions.xml</code>	Contains the regular expressions that define individual tokens in a COSPAS-SARSAT field with a reference to the error reported should a parsing error occur;
<code>COSPAS_Output_Document.xsd</code>	The XML Schema file describing the W3C output document;
<code>COSPAS_Error_Text.xml</code>	Contains errors reported by the parser;

**Table 3 - Descriptor Files Overview**

The relationship between these files is shown in the following diagram.



**Figure 2 - Descriptor Files Relationship**

### 2.7.1 Supported Messages File

This file specifies the supported COSPAS-SARSAT messages and their associated message field content. Nested message fields can be defined that provides a mechanism to specify message fields that occur multiple times.

Each of the message fields specified must have a corresponding entry in the **COSPAS\_Field\_Definitions.xml** descriptor file.

The order in which the fields appear is important and must follow the sequence specified in the COSPAS-SARSAT specification given in [1].

Node Name	Attributes	Description	Valid Values
SIT	sitNumber	The SIT number for a COSPAS-SARSAT message.	Integer value, SIT number
MF_ID	None	The value of this tag specifies a field number that is part of the message defined by the SIT tag. This number must exist as a defined field in the field definition file.	Integer value, field number
MFrepeatBlock	fieldOccurrences	<p>This tag provides a mechanism to define recursive nested blocks of MF_ID tags that may occur 'n' times, but the actual number of times the blocks occur is specified at run time with a value contained in another field that is specified in this attribute.</p> <p>As an example, assume that the MFrepeatBlock contains 3 MF_ID tags. The MFrepeatBlock fieldOccurrences value is '8'. This means that the block of 3 MF_ID fields will be repeated in a message the number of times specified in field 8.</p> <p>The MFrepeatBlock tags can be nested; the current implementation is limited to three levels of nesting. However, it should be noted that the parser software is able to deal with an infinite number of nesting levels (limited only by hardware of course); the limit is imposed using an XSD file for this configuration file, which limits the nested levels to 3. This is easily modified should more nested levels be required.</p>	Integer value, field number

**Table 4 - Supported Messages File Nodes and Attributes**

## 2.7.2 Field Definition File

This file contains a definition for each of the COSPAS-SARSAT message fields. It does this by defining each field with a number as a 'key' to the field and for each field, a state machine, which effectively defines the order in which tokens in a field must occur. For each machine node a regular expression is given that is used to match a field token to check if it is syntactically and semantic correct. The regular expression must exist in the **COSPAS\_Field\_Definitions.xml** file.

Node Name	Attributes	Description	Valid Values
MF	MFnumber	The COSPAS-SARSAT field number as specified in [1]. The field number is used as a key from the supported message file and must be a valid COSPAS-SARSAT field number.	Integer, field identifier
	MFname	The COSPAS-SARSAT field name as given in the specification.	String
machine	number	A state machine number; there must be at least one machine node with the number '1'. The number '1' machine is considered the start machine by the software implementing the state machine. All other machines can be numbered with any positive integer value.	Positive Integer
node	nextMachineNode	<p>This tag can occur 'n' times within a machine tag. Each machine node contains a reference to a regular expression used to match a field token. The list of nodes is a mechanism to allow the state machine to be exited to any other state machine depending on which of the node regular expressions matches the token being processed.</p> <p>For example, assume a token is being processed and there are three node tags each with a regular expression; the software compares the token with each of the regular expressions in turn, until a match is found. Once a match is found, the node tag contains the 'nextMachineNode' that specifies the next machine to jump to.</p> <p>If no match is found an error is reported as specified in the regular expression file.</p>	Positive Integer, the number must have a corresponding MF number attribute defined.

**Table 5 - Field Definition File Nodes and Attributes**

### 2.7.3 Sub-fields

This file specifies the regular expressions for any tokens that may occur in a COSPAS-SARSAT message. The expressions have been constructed from information provided in [1] describing the field syntax and semantics.

For each regular expression, there is an associated error message reference used to look up an error message that is reported should a regular expression fail to match with the expression.

The order in which the fields appear is unimportant, although for clarity it is recommended that they be entered in alphabetical order to make it easier to locate a field.

Node Name	Attributes	Description	Valid Values
REGEXP	regExpNameNone	The value of the REGEXP is a valid regular expression. The attribute is an identifying name used by the state machine as an index to look up a regular expression	String for the regular expression String for the regular expression
	errorTextNumber	An error number used as an index into the error text file.	Integer, must exist in the error text file.

**Table 6 - Regular Expression File Nodes and Attributes**

### 2.7.4 Error Text File

This file contains the error messages reported by the parser.

The nodes and attributes in the auxiliary term file are described in the following table.

Node Name	Attributes	Description	Valid Values
ERROR_TEXT	errNum	The value of the tag is the error text that will be reported. The attribute is an integer used as an index to reference an error message.	String for the error text; Integer to index the error message;

**Table 7 - Error Text File Node and Attribute Descriptions**

## 2.8 W3C XML Output Document Structure

The software is provided with an XSD document that describes, in detail, the structure and content of the W3C XML output document.

The overall structure is as shown below:

```
<COSPAS_Output_Document>
  <Fields SITnumber="121">
    <Field fieldNumber="1" startIdx="0" endIdx="15">FIELD DATA</Field>
    <Field fieldNumber="2" startIdx="10" endIdx="25">FIELD DATA</Field>
    <Field fieldNumber="3" startIdx="20" endIdx="35">FIELD DATA</Field>
    <Field fieldNumber="4" startIdx="30" endIdx="45">FIELD DATA</Field>
  </Fields>
  <Errors>
    <Error startIdx="0" endIdx="30" fieldNumber="12" token="4">ERROR TEXT</Error>
    <Error startIdx="0" endIdx="30" fieldNumber="12" token="4">ERROR TEXT</Error>
    <Error startIdx="0" endIdx="30" fieldNumber="12" token="4">ERROR TEXT</Error>
    <Error startIdx="0" endIdx="30" fieldNumber="12" token="4">ERROR TEXT</Error>
  </Errors>
  <Original_Message>MESSAGE GOES IN HERE</Original_Message>
  <Information>
    <Info>WILL ADD THINGS AS REQUIRED IN HERE</Info>
  </Information>
</COSPAS_Output_Document>
```

**Figure 3 - W3C Output Document File Structure**

The node names shown in the figure are described in the following table. One way with which to access the data in the W3C output document would be to use Xpath with the path names as described in the XSD.

Node Name	Attribute	Description
Fields	SITnumber	The SIT number for which this output file pertains.
Field	fieldNumber	The value of this tag is the field as extracted from the original message. The field number to which this tag pertains
	startIdx	The zero based index where this field starts in the original message;
	endIdx	The zero based index where this field ends in the original message;
Errors	fieldNumber	The value of this tag is the error text reported for the token identified by the start and end index provided in this tags attributes. The field number to which this tag pertains

Node Name	Attribute	Description
	startIdx	The zero based index where the erroneous token starts in the original message;
	endIdx	The zero based index where the erroneous token ends in the original message;
	token	The erroneous token
Original_Message		The original message
Information	Info	Used for miscellaneous data; currently reserved for future use.

**Table 8 - Output File Node and Attribute Descriptions**

### 2.8.1 Accessing the Output Data

This section provides guidance on how to retrieve data from the W3C document and some additional caveats to be observed when accessing the output document.

If a field is not present in a received message then the corresponding node will be missing from the W3C output document. Hence node retrieval can return null and should be allowed for in the code accessing the W3C output document.

One method to access data in the W3C output document is to use Xpath queries.

As the 'Field' and 'Error' nodes may appear more than once, when retrieving data from those fields containing multiple nodes, a 'Node List' must be recovered and iterated over to retrieve the individual child nodes.

## 3 The API

The API for the parser is very simple with a single public method available:

- ***parseMessage***– Parses a COSPAS-SARSAT message;

When instantiating the parser, the message to parse is passed into the constructor along with an offset value indicating where a message starts with respect to a complete message that includes an AFTN message header.

### 3.1 Method Synopsis

#### 3.1.1 Parser Instantiation

When instantiating the parser, the message to parse must be passed into the constructor. In addition there is an offset index, which must be set to the index of the first COSPAS-SARSAT message, (the '/' or '1.') if the message errors and field indices are to be based on the complete message including the AFTN header. If this is not a concern, the offset can be set to zero.

##### Method Signature

```
public COSPAS_Parser(final String message, int offset)
```

##### Input Parameters

message - A string containing the message;

offset – a zero based index to the start of the COSPAS-SARSAT message with reference to the start of an AFTN message that includes the AFTN header;

##### Output Parameters

None;

#### 3.1.2 Parse Message

This method parses a COSPAS-SARSAT message excluding the AFTN message header. The message is expected to start with a forward slash '/' or '1. ', (SIT 185).

This method returns a W3C XML document containing the fields of a message as described in the XSD. Refer to the XSD document for a description of the schema and other notes in this document.

Any errors that are detected are included in the XML document.

It is the caller's responsibility to test for errors in the XML output document by examining if the 'Errors' node is empty or not.

##### Method Signature

```
public org.w3c.dom.Document parseMessage()
```

##### Input Parameters

None;

##### Output Parameters

The W3C XML document containing the parsed message;

## 4 Appendix A – Acronyms

	Acronym	Description
<b>A</b>		
	ADD	Airport Data Display
	AFTN	Aeronautical Fixed Telecommunications Network
	API	Application Programmers Interface
	ATM	Air Traffic Management
<b>B</b>		
<b>C</b>		
	COSPAS	Cosmitscheskaja Sistema Poiska Awarinitsch Sudow (Russian: space system for search of vessels in distress)
<b>D</b>		
<b>E</b>		
<b>F</b>		
<b>G</b>		
<b>H</b>		
<b>I</b>		
	ID	Identifier
<b>J</b>		
	JRE	Java Run Time Environment
<b>L</b>		
	LUT	?
<b>M</b>		
	MCC	Mission Control Center
	MF	Message Field
	MHz	Mega Hertz
<b>N</b>		
<b>O</b>		
<b>P</b>		
<b>R</b>		
<b>S</b>		
	SARP	Search And Rescue Processor
	SARR	Search and Rescue Repeater
	SARSAT	Search And Rescue Satellite Aided Tracking
	SIT	Satellite Interactive Terminal
	SPOC	Space Program Operations Contract (NASA)
<b>T</b>		
	TM	Trajectory Management
<b>U</b>		
	UTF	Unicode Transformation Format (16 bit Unicode to 7/8 bit character conversion)



	<b>Acronym</b>	<b>Description</b>
<b>V</b>		
<b>W</b>		
	W3C	World Wide Web Consortium
<b>X</b>		
	XML	Extensible Mark up Language
	XSD	XML Schema Definition

**Table 9 - Acronyms**

## 5 Appendix D – XSD For W3C Output Document

This section contains the XSD used to describe the W3C output document produced by the Flight ATM Systems Ltd. COSPAS-SARSAT Parser.

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!--Copyright 2015 Flight ATM Systems Ltd.
```

```
This file and its contents both textual and intellectual remains the  
sole and exclusive property of Flight ATM Systems Ltd. No part of it  
may be copied, or disclosed by the recipient to third persons, without  
the prior written consent of Flight ATM Systems Ltd.; nor shall it be  
used for any purpose other than in connection with an agreement or  
proposed agreement with Flight ATM Systems Ltd.
```

```
Flight ATM Systems Ltd. Registration number 5625816 * * http://www.flightatm.com -->
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="COSPAS_Output_Document">
    <xs:complexType>
      <xs:sequence>

        <xs:element name="Fields">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Field" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="fieldNumber" type="xs:int"/></xs:attribute>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="SITnumber" type="xs:int"/></xs:attribute>
          </xs:complexType>
        </xs:element>

        <xs:element name="Errors">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Error" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="startIdx" type="xs:int"/></xs:attribute>
                  <xs:attribute name="endIdx" type="xs:int"/></xs:attribute>
                  <xs:attribute name="fieldNumber" type="xs:int"/></xs:attribute>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
        <xs:attribute name="token" type="xs:int"/></xs:attribute>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="Original_Message" type="xs:string"/></xs:element>

<xs:element name="Information">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Info" type="xs:string"/></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```