

```
<!DOCTYPE ichicsrack [  
<!-- PUBLIC "-//ICHM2//DTD ICH ICSR Acknowledgment Vers. 1.1//EN" "ich-icsrack-  
v1.1.dtd" -->
```

```
<!--
```

Individual Case Safety Report Acknowledgment Document Type Definition

The DTD issued by the ICH M2 group and is public domain in nature.

No one can claim copyright on this DTD.
No commercial distribution is allowed.

The ICH is not responsible for any damage or financial loss resulting from use of this DTD. This version is tentative in nature and changes are expected. This DTD is subject to the ICH M2 change control procedures.

Version 1.0, Release 1.0
Version 1.1, Release 1.0, 10 November 2000

```
-->
```

```
<!-- TECHNICAL NOTE
```

ICH ICSR SGML Declaration

To correctly parse an ICH ICSR SGML message requires the selection of the correct SGML declaration, along with this DTD, and the ICH ICSR SGML instance. This technical note provides guidance on which of several SGML declarations included with the ICH ICSR application to select based on the language and character set used within the ICH ICSR SGML instance. The method by which an SGML parser is told to use a specific declaration is parser specific. The three most common methods are:

1. Tell the parser via the command line which declaration to use.
2. Tell the parser via a specific environment variable which declaration to use.
3. If the parser supports SGML Open catalogs, within the catalog file is a command that can tell the parser which declaration to use. SGML catalog files are an industry standard way to tell a parser how to find all of the pieces (declaration, DTD, SGML document instance) needed to successfully parse.

To cover all of the languages necessary to support the ICH ICSR application, ISO 10646 (UNICODE) would have to be used. If this were the case, then only one SGML declaration would be needed to support all ICH ICSR languages. Because UNICODE support is not available in all popular computer application programs, the ICH M2 EWG has recommended that a variety of character sets be used at this time, instead of a single UNICODE character encoding. For each of these different character encodings a different SGML declaration needs to be generated and used.

Five SGML declarations are included with the ICH ICSR distribution. They are described briefly below.

ich-icsr-latin1.dcl - This SGML declaration supports the ISO 8859-1 (Latin 1) character set. This character set supports English, and most Western European languages.

ich-icsr-latin7.dcl - This SGML declaration supports the ISO 8859-7 (Latin 7) character set. This character set supports Greek.

ich-icsr-sjis.dcl - This SGML declaration supports the Shift JIS character set for encoding Japanese.

ich-icsr-utf8.dcl - This SGML declaration supports the ISO 10646 (UNICODE) UTF-8 character set. This character set supports almost all of the worlds currently written languages.

ich-icsr-mult.dcl - This SGML declaration will support all of the currently defined ICHICSR languages, and it doesn't require the use of UNICODE. It is however a "hack" that relies on a fortuitous characteristic of the languages currently being used within the ICH ICSR SGML application.

This hack works because the character set documented within an SGML declaration is for the benefit of both the SGML parser, and the programmer who implements the SGML application. The programmer needs detailed knowledge of all the characters being used to correctly code an application. The parser only really needs to about some of the characters. In particular it needs to know which characters signify an SGML event. For instance that the "<" character starts an SGML open tag, and the "</" character sequence starts an SGML close tag. All told, the parser needs to know very little. Because of the particular combination of character sets being used within the ICH ICSR application, an SGML declaration can be defined that tells the parser just what it needs to know and still work across all of the different character sets. The catch is that this SGML declaration doesn't tell the programmer all that she or he needs to know to correctly code the rest of the application. If this declaration is used, the programmer will have to get this detailed information from another source.

-->

```
<!-- ===== -->
<!-- Entities -->
<!-- ===== -->
```

```
<!-- Use the lang attribute to indicate the language of an --
-- elements content via an ISO 639 language Code. -->
```

```
<!ENTITY % lang.att "lang CDATA #IMPLIED">
```

```
<!-- Standard Character Entities to escape SGML special characters. --
-- When "<", ">", and "&" occur in text, they should be replaced --
-- by "&lt;", "&gt;", and "&amp;" respectfully. -->
```

```
<!-- Less Than "<" -->
<!ENTITY lt "&#38;#60;">
```

```
<!-- Greater Than ">" -->
<!ENTITY gt "&#62;">
```

```
<!-- Ampersand "&" -->
```

```

<!ENTITY amp      "&#38;#38;">

<!-- ===== -->
<!-- Elements and Attributes -->
<!-- ===== -->

<!ELEMENT ichicsrack      - -
      (ichicsrmessageheader ,
       acknowledgment)>
<!ATTLIST ichicsrack
      lang CDATA #REQUIRED
>

<!-- M.1 ICH ICSR Message Header Information -->

<!ELEMENT ichicsrmessageheader      - -
      (messagetype
       messageformatversion
       messageformatrelease
       messagenumb
       messagesenderidentifier
       messagereceiveridentifier
       messagedateformat
       messagedate)>
<!ATTLIST ichicsrmessageheader
      %lang.att;
>

<!-- M.1.1 Message Type -->
<!ELEMENT messagetype      - - (#PCDATA) >
<!ATTLIST messagetype
      %lang.att;
>

<!-- M.1.2 Message Format Version -->
<!ELEMENT messageformatversion      - - (#PCDATA) >
<!ATTLIST messageformatversion
      %lang.att;
>

<!-- M.1.3 Message Format Release -->
<!ELEMENT messageformatrelease      - - (#PCDATA) >
<!ATTLIST messageformatrelease
      %lang.att;
>

<!-- M.1.4 Message Number -->
<!ELEMENT messagenumb      - - (#PCDATA) >
<!ATTLIST messagenumb
      %lang.att;
>

<!-- M.1.5 Message Sender Identifier -->
<!ELEMENT messagesenderidentifier      - - (#PCDATA) >
<!ATTLIST messagesenderidentifier
      %lang.att;
>

```

```

<!-- M.1.6 Message Receiver Identifier -->
<!ELEMENT messagereceiveridentifier - - (#PCDATA) >
<!ATTLIST messagereceiveridentifier
    %lang.att;
>

<!-- M.1.7a Message Date Format -->
<!ELEMENT messagedateformat - - (#PCDATA)>
<!ATTLIST messagedateformat
    %lang.att;
>

<!-- M.1.7b Message Date -->
<!ELEMENT messagedate - - (#PCDATA) >
<!ATTLIST messagedate
    %lang.att;
>

<!-- A.1 ICSR Acknowledgment Message-->
<!ELEMENT acknowledgment - -
    (messageacknowledgment
    ,
    reportacknowledgment*)>
<!ATTLIST acknowledgment
    %lang.att;
>

<!-- A.1 Message Acknowledgment -->
<!ELEMENT messageacknowledgment - -
    (icsrmessagenumb
    ,
    localmessagenumb?
    ,
    icsrmessagesenderidentifier
    ,
    icsrmessagereceiveridentifier
    ,
    icsrmessagedateformat
    ,
    icsrmessagedate
    ,
    transmissionacknowledgmentcode
    ,
    parsingerrormessage?)>
<!ATTLIST messageacknowledgment
    %lang.att;
>

<!-- A.1.1 ICSR Message Number -->
<!ELEMENT icsrmessagenumb - - (#PCDATA)>
<!ATTLIST icsrmessagenumb
    %lang.att;
>

<!-- A.1.2 Local Message Number -->
<!ELEMENT localmessagenumb - - (#PCDATA)>
<!ATTLIST localmessagenumb
    %lang.att;
>

<!-- A.1.3 ICSR Message Sender Identifier -->
<!ELEMENT icsrmessagesenderidentifier - - (#PCDATA)>
<!ATTLIST icsrmessagesenderidentifier
    %lang.att;

```

```

>

<!-- A.1.4 ICSR Message Receiver Identifier -->
<!ELEMENT icsrmessagereceiveridentifier - - (#PCDATA)>
<!ATTLIST icsrmessagereceiveridentifier
    %lang.att;
>

<!-- A.1.5a ICSR Message Date Format -->
<!ELEMENT icsrmessagedateformat - - (#PCDATA)>
<!ATTLIST icsrmessagedateformat
    %lang.att;
>

<!-- A.1.5b ICSR Message Date -->
<!ELEMENT icsrmessagedate - - (#PCDATA)>
<!ATTLIST icsrmessagedate
    %lang.att;
>

<!-- A.1.6 Transmission Acknowledgment -->
<!ELEMENT transmissionacknowledgmentcode - - (#PCDATA)>
<!ATTLIST transmissionacknowledgmentcode
    %lang.att;
>

<!-- A.1.7 Parsing Error Message -->
<!ELEMENT parsingerrormessage - - (#PCDATA)>
<!ATTLIST parsingerrormessage
    %lang.att;
>

<!-- B.1 Report Acknowledgment -->
<!ELEMENT reportacknowledgment - -
    (safetyreportid
      safetyreportversion?
      localreportnumb?
      authoritynumb?
      companynumb?
      receiptdateformat?
      receiptdate?
      reportacknowledgmentcode
      errormessagecomment?)>
<!ATTLIST reportacknowledgment
    %lang.att;
>

<!-- B.1.1 Safety Report Identification -->
<!ELEMENT safetyreportid - - (#PCDATA)>
<!ATTLIST safetyreportid
    %lang.att;
>

<!-- B.1.2 Safety Report Version -->
<!ELEMENT safetyreportversion - - (#PCDATA)>
<!ATTLIST safetyreportversion
    %lang.att;

```

```
>

<!-- B.1.3 Local Report Number -->
<!ELEMENT localreportnumb - - (#PCDATA)>
<!ATTLIST localreportnumb
    %lang.att;
>

<!-- B.1.4 Regulatory authority's case report number (E2BM - A.1.10.1)-->
<!ELEMENT authoritynumb - - (#PCDATA)>
<!ATTLIST authoritynumb
    %lang.att;
>

<!-- B.1.5 Other sender's case report number (E2BM - A.1.10.2) -->
<!ELEMENT companynumb - - (#PCDATA)>
<!ATTLIST companynumb
    %lang.att;
>

<!-- B.1.7a Receipt Date Format of the most recent information for this report
(E2BM - A.1.7) -->
<!ELEMENT receiptdateformat - - (#PCDATA)>
<!ATTLIST receiptdateformat
    %lang.att;
>

<!-- B.1.7b Date of receipt of the most recent information for this report (E2BM
- A.1.7) -->
<!ELEMENT receiptdate - - (#PCDATA)>
<!ATTLIST receiptdate
    %lang.att;
>

<!-- B.1.8 Acknowledgment Code for a report -->
<!ELEMENT reportacknowledgmentcode - - (#PCDATA)>
<!ATTLIST reportacknowledgmentcode
    %lang.att;
>

<!-- B.1.9 Error Message or Comment -->
<!ELEMENT errormessagecomment - - (#PCDATA)>
<!ATTLIST errormessagecomment
    %lang.att;
>
]>
```