

1 Overview

The TMC desires improvements to the liaison statement tracking capabilities in the datatracker to address community requests and to simplify ongoing maintenance of the related code.

References to ticket nnn are to <http://trac.tools.ietf.org/tools/ietfdb/ticket/nnn>.

2 Deliverables / Tasks

- 2.1 Migrate the codebase and data to new models, refining the models as necessary.
- 2.2 Add history to liaison statements, capturing when they were received, posted, resent, approved, marked dead or modified.
- 2.3 Add Point of Contact email addresses for each body to be used as the default Response Contact.
- 2.4 Capture other SDOs identifiers for liaison statements.
- 2.5 Add "Action Holder" email addresses to incoming liaison statements.
- 2.6 Add the ability for a liaison statement to reference multiple other liaison statements as related.
- 2.7 Allow a liaison statement to be sent from multiple bodies and sent to multiple bodies.
- 2.8 Add the concept of a "dead" liaison statement, allowing an approver or the secretariat to remove a statement from the pending queue without posting it. Add a view to manage the set of dead liaison statements.
- 2.9 Add the ability to resend a liaison statement.
- 2.10 Improve search over posted liaison statements, searching all fields and allowing the results to be limited to a particular source and/or destination body and/or a particular time range. Facilitate browsing a series of related liaison statements as a "thread".
- 2.11 Address known issues with the liaison statement entry form when used as the secretariat and when used as a liaison manager.

- 2.12 Make it easier to see which liaison statements still need action.
- 2.13 Ensure liaison statement approvers are notified when there is something to approve.
- 2.14 Remove the restriction against duplicate liaison statement titles.
- 2.15 Allow the set of attachments to a liaison statement to be edited. In particular allow an attachment to be deleted. Leave a record in the history for each editing event.
- 2.16 Improve test coverage, and ensure tests exercise the use cases detailed in this document.

3 Detailed Description

3.1 Process background

Liaison relationships with other standards setting bodies are managed by the IAB. The processes for managing the relationships are detailed in RFC4052, and the procedures for handling liaison statements are detailed in RFC4053. The current liaison statement management tool (LSMT) began from the definitions in that RFC and the LSMT has been improved through previous contracted development. Individuals using the tool must follow the guidelines in RFC4691. Developers working on the liaison statement management section of the datatracker must be familiar with all three RFCs.

3.2 Use cases

3.2.1 Sending Liaison statements

In each of the below use cases, the secretariat can also enter the statement on behalf of the sender.

3.2.1.1 *Sent by the final approver*

When a statement is sent by an individual that has final approval authority (such as the IETF chair sending a statement on behalf of the IETF):

1. The statement is entered as a new outgoing liaison statement
2. The statement is emailed by the datatracker to the addresses indicated in the statement
3. The statement appears in the list of posted liaison statements

Alternatively, the statement may have been sent by the individual using their own email client, in which case the user will ask only to “Post” the liaison statement instead of “Post and Send”, in which case step 2 is skipped.

3.2.1.2 Sent by someone who is not the final approver, but has the necessary approval

When a statement is sent by an individual that does not have final approval authority (such as a working group chair sending a statement on behalf of their working group, which requires Area Director approval):

1. The statement is entered as a new outgoing liaison statement, capturing that approval has already been obtained
2. Steps 2 and 3 from section 3.2.1.1

3.2.1.3 Sent by someone who is not the final approver, and does not have the necessary approval yet

1. The statement is entered as a new outgoing liaison statement, capturing that approval has not been obtained
2. The statement enters the pending queue
3. The datatracker sends the approver email noting approval is requested
4. The approver views statements in the pending queue
5. The approver makes any appropriate edits to the liaison statement
6. The approver approves the statement
7. Steps 2 and 3 from section 3.2.1.1

Alternatively, if the approver disapproves the statement, the statement enters the “dead” state and no longer appears in the pending queue, and is not added to the list of posted statements.

3.2.2 Incoming Liaison statements

3.2.2.1 Statement is posted by secretariat or liaison manager

When an incoming liaison has been sent via email by another organization, the liaison manager (or the secretariat operating on their behalf) can post the statement to the datatracker:

1. The statement is entered as a new incoming liaison statement, and saved by pressing “Post” (not “Send and Post”)
2. The statement appears in the list of posted liaison statements

3.2.2.2 Statement is sent only to the secretariat

Occasionally an incoming liaison statement is sent directly and only to the secretariat.

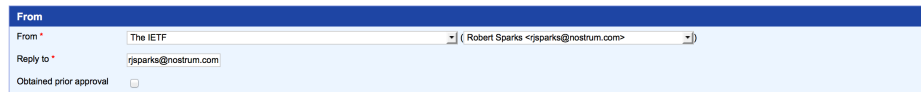
1. The statement is entered as a new incoming liaison statement, and saved by pressing “Send and Post”
2. The statement is emailed to the To and CC email addresses
3. The statement appears in the list of posted liaison statements

3.3 General Expectations

This project is a set of incremental improvements to the existing LSMT. Changes to the user interface are expected to be the minimum required to provide the improved functionality. The general look and feel of the tool is to be preserved.

This project will correct the user interface to consistently say “Liaison Statement” and not “Liaison”.

3.4 Changes for the secretariat



In the above subsection of the new liaison statement form, the person part of the “From” field will be changed to be an edit control instead of a pull-down list. The control will autocomplete similar to the control for editing the shepherd of a document.

The “Reply to” field (which will be relabeled – see section 3.6) will be populated with the new point of contact address(es) for the selected “From” body.

A link will be added near the “From” field to easily access the page for creating a new group at </secr/groups/add/>.

Currently (see ticket 737), when the “To” field is changed on the new liaison statement form, the “From” field is reset. This project will remove that coupling.

3.5 Changes for approvers

For each liaison statement shown at /liaison/for_approval/, the detail page at (for example) [liaison/for_approval/1302/](/liaison/for_approval/1302/) will include a new “Mark Dead” button in addition to the existing “Approve” button. A new view will be created to list the set of liaison statements that have been marked dead at </liaison/dead/>, which will have exactly the same access permissions as /liaison/for_approval. The new view will allow a dead liaison statement to be “revived”, placing it back into the pending queue.

3.6 Changes for liaison managers

This project will add the ability to resend any liaison statement that has been sent and posted. The resending will be noted in the history for the liaison statement. See ticket 678.

This project will allow a liaison statement to be sent from multiple bodies, to multiple bodies. See ticket 726. The current concept of associating a Person with the “From” body will be extended to associate a different person with each indicated “From” body (see Section 3.4). The code will not attempt to prevent nonsensical combinations of “To” and “From” bodies.

This project will ensure that approvers get an email notice from the datatracker whenever a liaison statement is entered that needs their approval. See tickets 879 and 880.

This project will add a field to capture any identifiers that other bodies use for a liaison statement. Note that this may be multiple identifiers since a liaison statement may come from more than one body. See ticket 225.

This project will allow a new liaison statement to be associated with arbitrarily many existing liaison statements as “related”. See ticket 739.

This project will allow a liaison manager or the secretariat to add new attachments to an existing liaison statement, edit the names of existing attachments, and to remove attachments from a statement. Each change will be captured as events in the statement’s history. See ticket 738. Removed attachments will not be deleted from disk, and will remain associated with the liaison statement in the database. Removed attachments will be listed on the attachment editing page, clearly identified as removed, and may be unremoved (reattached). A removed attachment must not appear in any other view.

This project will add a field to the model for liaison statements to capture an “action holder” contact (a Person) for incoming liaison statements. See ticket 736.

This project will add a default set of CC addresses for each organization that can be maintained from the django admin. See tickets 727 and 730. These lists will initially be populated to match what’s currently coded, with the following additions:

- itu-t-liaison@iab.org will be added to each of these ITU related groups
 - ITU-T
 - ITU-T NGN
 - ITU-T SG 15
 - ITU-T MPLS
 - ITU-T SG 16
 - ITU-T SG 17
 - ITU-T SG 11

- ITU-T SG 2
- ITU-T SG 3
- ITU-T SG15, Q9, Q10, Q12 and Q14
- ITU-T SG 13
- ITU-T TSAG
- ITU-T SG 12
- ITU-T Working Party 5/13
- ITU-T FG Distraction
- ITU-T JCA-Cloud
- ITU-T JCA-COP
- ITU-R
- ITU-T Study Group 12, Q17/12
- The area directors will be added to the list for working groups

This project will remove the current restriction against duplicate liaison statement titles.

This project will replace the current “Reply To” and “Response Contact” fields with a single “Response Contact” field. The entry form will show that field where “Reply To” is currently shown.

This project will add tool-tip definitions to the fields on the liaison statement entry form. The meanings of “Response Contact” and “Technical Contact” have been particularly hard for new liaison managers to understand. The UI should unobtrusively define those two fields directly on the entry page. See ticket 725. The project will also update the template and CSS used by the form to remove dead class and identifiers and ensure that all fields are shown with appropriately sized controls. See ticket 733.

This project will add a verification that when an outgoing liaison statement is sent “In Response” that one or more other liaison statements are linked as related. That is, it is an error to send a liaison statement “In Response” without capturing what is being responded to.

This project will improve the attachment subsection of the liaison statement entry form. Currently it is too easy to save the liaison statement without saving the attachments. The form should, at a minimum, warn if there are unsaved attachments selected.

The current liaison statement entry form always defaults the “To” field to 3GPP even if the logged in user cannot send a liaison statement to 3GPP. This project will set the initial value to the first body to which the logged in user can actually send a statement.

3.7 Changes to liaison statement lists and search results

Currently, posted liaison statements are only accessible in one list at <http://datatracker.ietf.org/liaison/>. This project will add the ability to search over the set of posted liaison statements. The search form will provide text search, optionally limited by source and/or destination body, and by date range.

- Input to the text search will consist of a single UTF-8 input string that will be matched case-insensitive against either all the text fields in the liaison statement (including the titles of attachments), or against just the title field (as indicated by the user). The search will not compare against the content of attachments.
- Results can be limited to a single source and/or single destination body. Support for limiting searches to multiple from or to bodies is not required at this time.
- Results can be limited to starting and/or ending at given dates.

The full list will be displayed the same as a search result that matches all records.

Liaison statements in any state other than posted (in particular those in the pending queue) will not be included in search results.

The resulting display will make those liaison statements that are awaiting action (see RFC4053 section 2.4 and ticket 740) visually distinctive.

The result list will be sortable by any of the displayed columns.

The combination of searching by source and destination bodies, limiting by date range, and being able to sort by time will provide the ability to see a "thread" of statements between two dialogs listed. That thread will be readable either by navigating sequentially from these search results, or by following the related statement links from the views of the individual liaison statements.

3.8 Changes to the view of a given liaison statement

This project will make it easy to navigate between related liaison statements.

3.9 Implementation Details

Previous work on the LSMT resulted in the creation of the IETFHeirarchyManager in `liaisons/utls.py`. Experience has shown that the IETFHeirarchyManager is unnecessarily complex. This project will remove IETFHeirarchyManager from the codebase, replacing its functionality using the 'liaiman' Roles that are already populated in the datatracker.

3.9.1 New Models

```
class LiaisonStatementState(NameModel):
    "Pending, Approved, Dead"
    pass
```

```

class LiaisonStatementEventTypeName(NameModel):
    "Submitted, Modified, Approved, Posted, Killed, Resurrected, MsgIn,
    MsgOut, Comment"
    pass

class LiaisonStatementTagName(NameModel):
    "Action Required, Action Taken"
    pass

class LiaisonStatement(models.Model):
    title = models.CharField(blank=True, max_length=255)
    other_identifiers = models.TextField() # Identifiers from other bodies
    purpose = models.ForeignKey(LiaisonStatementPurposeName)
    body = models.TextField(blank=True)
    deadline = models.DateField(null=True, blank=True)

    from_groups = models.ManyToManyField(Group, blank=True)
    from_name = models.CharField(max_length=255, help_text="Name of the
    sending bodies")

    to_groups = models.ManyToManyField(Group, blank=True)
    to_name = models.CharField(max_length=255, help_text="Name of the
    recipient bodies")

    from_contact = models.ForeignKey(Email, blank=True,
    null=True)
    to_contacts = models.CharField(blank=True, max_length=255,
    help_text="Contacts at recipient body")
    response_contacts = models.CharField(blank=True, max_length=255,
    help_text="Where to send a response") # RFC4053
    technical_contacts = models.CharField(blank=True, max_length=255,
    help_text="Who to contact for clarification") # RFC4053
    action_holder_contacts = models.CharField(blank=True, max_length=255,
    help_text="Who makes sure action is completed") # incoming only?
    cc_contacts = models.TextField(blank=True)

    # Removable attachments
    attachments = models.ManyToManyField('Document',
    through='LiaisonStatementAttachments', blank=True)

class RelatedLiaisonStatement(models.Model):
    source = models.ForeignKey(LiaisonStatement)
    target = models.ForeignKey(LiaisonStatement)
    relationship = models.ForeignKey(DocRelationshipName) # make new name
    type if necessary

class LiaisonStatementAttachments(models.Model):
    statement = models.ForeignKey(LiaisonStatement)
    document = models.ForeignKey(Document)
    removed = models.BooleanField()

class LiaisonStatementGroupContacts(models.Model):
    group = models.ForeignKey(Group, unique=True)
    default_reply_to = models.CharField(max_length=255)

class LiaisonStatementEvent(models.Model):
    time = models.DateTimeField(auto_now_add=True)
    type = models.ForeignKey(LiaisonStatementEventTypeName)
    by = models.ForeignKey(Person)
    statement = models.ForeignKey(LiaisonStatement)
    desc = models.TextField()
    msg = models.ForeignKey(Message, null=True, blank=True)

```


3.9.2 Migrate legacy data to new models

There are 1226 LiaisonStatements to migrate.

Events will have to be synthesized from the current submitted, modified, and approved fields.

Legacy statements will have 0 or 1 RelatedLiaisonStatement object, depending on the current related_to field.

The to_contact and response_contact fields from the current model will be combined into the response_contacts field in the new model. The migration will add a comment event capturing the original field contents.

The data migration will ensure that to_groups and from_groups always reference at least one Group. In the current data, 484 statements have an empty to_group, and 245 statements have an empty from_group. For those statements, the UI currently displays the to and from groups using the from_name and to_name fields. The migration will determine the appropriate Group objects to point to based on those fields, which will require some manual inspection. Typical from_name and to_name entries look like "SIP", "The IETF", or "The IESG and the IAB". More complicated entries look like "SG17, SG13, SG11, JCA-NID, ETSI TISPAN WG4, 3GPP TSG CT4, IESG". After migration, from_name and to_name are expected to primarily be empty, only having content for exceptional cases. The 'name' function in the LiaisonStatement class will be changed to return the contents of to_name and from_name when they are not empty.

Robert Sparks 2/7/14 2:37 PM

Comment: Right now, the display code shows the titles of the attachments when there is no title for the LS. See for example <http://datatracker.ietf.org/liaison> on 2010-10-29 for the statement from MPEG to the IETF.

4 Expected Development Processes and Practices

The contractor will adhere to the requirements at <http://trac.tools.ietf.org/tools/ietfdb/wiki/ContractorInstructions?version=11>