

Emacs SASL Library

by Daiki Ueno

Copyright © 2000 Daiki Ueno.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

この文書を、フリーソフトウェア財団発行の GNU フリー文書利用許諾契約書第 1.2 版またはそれ以降の版が定める条件の下で複製、配布、あるいは変更することを許可します。変更不可部分、表表紙テキストおよび裏表紙テキストは指定しません。この利用許諾契約書の複写は「GNU フリー文書利用許諾契約書」という章に含まれています。

Emacs SASL

このマニュアルは Emacs SASL ライブラリーを説明します。

異なるプロトコルを使うアプリケーション間で、複数の認証メカニズムを共有するための共通インターフェースです。

1 大要

SASL は *Simple Authentication and Security Layer* の略です。標準は RFC2222 に文書化されています。これは、さまざまなアプリケーション・プロトコルに、認証サポートを追加するための簡単な手段を提供します。

このライブラリーの最上位のインターフェースは、Java SASL Application Program Interface によって触発されました。それは、抽象化した一連の認証メカニズムのドライバーを定義します。

バックエンドのドライバーは、なるべく認証メカニズムに密接していることを目指しています。どこにある追加の設定情報にも、この実装からアクセスすることができます。

2 使い方

(まだ書かれていません)。

Emacs SASL ライブラリーを使うには、アプリケーション・プログラムの最初で、以下の式を評価して下さい。

```
(require 'sasl)
```

実行時にのみ `sasl.el` が存在していることを検査したいならば、代わりに必要な関数の `autoload` の設定を並べておいても構いません。

3 データの型

ネゴシエートされたセキュリティ層の伝送に使われる三つのデータの型として、メカニズム、クライアント・パラメーター、および認証ステップがあります。

3.1 メカニズム

メカニズム (`sasl-mechanism` オブジェクト) が、SASL 認証メカニズムのドライバーのスキームです。

`sasl-mechanisms` [Variable]
メカニズム名のリスト。

`sasl-find-mechanism mechanisms` [Function]
適切なメカニズムを取得します。
この関数は `mechanisms` と `sasl-mechanisms` を比較して、適切な `sasl-mechanism` オブジェクトを返します。

```
(let ((sasl-mechanisms '("CRAM-MD5" "DIGEST-MD5")))
  (setq mechanism (sasl-find-mechanism server-supported-mechanisms)))■
```

`sasl-mechanism-name mechanism` [Function]
メカニズムの名前を文字列で返します。

認証メカニズムのドライバー (Chapter 4 [Back end drivers], page 6) を書きたければ、`sasl-make-mechanism` を使い、`sasl-mechanisms` および `sasl-mechanism-alist` を正しく変更して下さい。

`sasl-make-mechanism name steps` [Function]
`sasl-mechanism` オブジェクトを割り当てます。
この関数は二つのパラメーター、すなわちメカニズムの名前、および認証するための関数のリストを受け取ります。

```
(defconst sasl-anonymous-steps
  '(identity ; 最初は応答無し
    sasl-anonymous-response))

(put 'sasl-anonymous 'sasl-mechanism
  (sasl-make-mechanism "ANONYMOUS" sasl-anonymous-steps))
```

3.2 クライアント

クライアント (`sasl-client` オブジェクト) は四つのパラメーター、すなわちメカニズム、ユーザー名、サービス名、およびサーバー名によって初期化されます。

`sasl-make-client mechanism name service server` [Function]
`sasl-client` オブジェクトを用意します。

`sasl-client-mechanism client` [Function]
クライアントのメカニズム (`sasl-mechanism` オブジェクト) を返します。

`sasl-client-name client` [Function]
クライアントの認証名を文字列で返します。

`sasl-client-service client` [Function]
クライアントのサービス名を文字列で返します。

`sasl-client-server client` [Function]
クライアントのサーバー名を文字列で返します。

属性の設定を追加したい場合は、`sasl-client-set-property` を使って下さい。

`sasl-client-set-property client property value` [Function]
与えられた属性およびその値をクライアントに追加します。

`sasl-client-property client property` [Function]
クライアントの属性の値を返します。

`sasl-client-set-properties client plist` [Function]
クライアントの属性を破壊的に設定します。
第二引数は新しいプロパティ・リスト (plist) です。

`sasl-client-properties client` [Function]
クライアントの設定のプロパティ・リストを、まるごと返します。

3.3 ステップ

ステップ (`sasl-step` オブジェクト) は認証「ステップ」の抽象化です。それは、応答の値と次の認証処理の入り口を保持します (後者はアクセスできません)。

`sasl-step-data step` [Function]
step が保持するデータを文字列で返します。

`sasl-step-set-data step data` [Function]
step に *data* の文字列を格納します。

最初の応答を得るには、第二引数を `nil` にして関数 `sasl-next-step` を呼ばなければなりません。

```
(setq name (sasl-mechanism-name mechanism))
```

この段階で、SASL の認証プロトコルの交換を開始するコマンドを送ることができます。例です。

```
(process-send-string
 process
 (if (sasl-step-data step)                ; 最初の応答
     (format "AUTH %s %s\r\n"
              name (base64-encode-string (sasl-step-data step) t))
     (format "AUTH %s\r\n" name)))
```

認証処理を続けるために行なわなければならないことは、連続して `sasl-next-step` を呼ぶことです。

`sasl-next-step client step` [Function]
認証ステップを実行します。
最初は *step* を `nil` に設定しなければなりません。

4 バックエンド・ドライバー

(まだ書かれていません)。

5 Index

(Index is nonexistent)

6 Function Index

sasl-client-mechanism.....	4	sasl-find-mechanism.....	4
sasl-client-name.....	4	sasl-make-client.....	4
sasl-client-properties.....	5	sasl-make-mechanism.....	4
sasl-client-property.....	5	sasl-mechanism-name.....	4
sasl-client-server.....	5	sasl-next-step.....	5
sasl-client-service.....	5	sasl-step-data.....	5
sasl-client-set-properties.....	5	sasl-step-set-data.....	5
sasl-client-set-property.....	5		

7 Variable Index

sasl-mechanisms	4
-----------------------	---

Short Contents

Emacs SASL	1
1 大要	2
2 使い方	3
3 データの型	4
4 バックエンド・ドライバ	6
5 Index	7
6 Function Index	8
7 Variable Index	9

Table of Contents

Emacs SASL	1
1 大要	2
2 使い方	3
3 データの型	4
3.1 メカニズム	4
3.2 クライアント	4
3.3 ステップ	5
4 バックエンド・ドライバー	6
5 Index	7
6 Function Index	8
7 Variable Index	9