

xls^v エージェント・リファレンス・マニュアル

森 洋久

joshua @ globalbase.org

2006-07-26 版

目次

第 1 章	はじめに	2
1.1	目的と概要	2
1.2	このマニュアルを読むために必要な知識	2
1.3	前提となるシステム要件	2
1.4	xlsv の XL プロトコル機能	2
1.5	xlsv の HTTP エージェント機能	3
第 2 章	xlsv 基本 XL 関数	5
2.1	概要	5
2.2	継承情報	5
2.3	関数	6
2.3.1	SetAgent (エージェントの起動)	6
2.3.2	HttpAgent (HTTP エージェント機能の設定)	7
2.3.3	DefineAgent (起動可能エージェントの設定)	8
第 3 章	HTTP エージェント呼び出し規定 (HTTP リクエスト関数)	9
3.1	概要	9
3.2	関数	10
3.2.1	HTTP (HTTP リクエスト関数用環境)	10
3.2.2	GET	11
3.2.3	POST	12
3.2.4	PUT	13
第 4 章	HTTP リクエスト関数の戻り値	14
4.1	概要	14
4.2	XML 要素	15
4.2.1	text/html	15
4.2.2	text/xml	16
4.2.3	image/jpeg	17
4.2.4	image/gif	18

第1章 はじめに

1.1 目的と概要

xlsv は、LANDSCAPE GLOBALBASE SERVER における、各種のパッシブコネクションを担当するエージェントです。XL プロトコルサーバとしての機能と、HTTP を受け、リクエストを GLOBALBASE のエージェント機能に翻訳する、HTTP エージェント機能を担当します。当該マニュアルでは、これらの機能の説明とリファレンスマニュアルを提供します。

1.2 このマニュアルを読むために必要な知識

このマニュアルを読む場合には、xl スクリプト [UNDEF REF (xl-lang)] の知識を必要とします。また、XL プロトコルに関しては、[UNDEF REF (xl-protocol)] を参照してください。また、GLOBALBASE の座標系の検索、重ね合わせの原理について概略をわかっている必要があります。これらの技術的理解については、GLOBALBASE 技術資料 [UNDEF REF (GLOBALBASE-tech)] に書かれています。手短に GLOBALBASE の機能を理解するためには、COSMOS GLOBALBASE VIEWER [1] を動かしてみることをお勧めします。

1.3 前提となるシステム要件

POSIX 系マシンが必要です。これに、LANDSCAPE GLOBALBASE VIEWER [2] をインストールすることによって xlsv の機能が利用可能となります。

1.4 xlsv の XL プロトコル機能

XL プロトコルは、TCP/IP、もしくはストリーム型のコネクション上に構築された、クライアントサーバ型のプロトコルです。クライアントからの要求、およびそれに対するサーバからの返答を、XML 形式、または、LISP 形式の書式でやりとりします。

ストリームの先がファイルであれば、ファイルから XML 形式のコマンドを受け取ることになり、また、ストリームの先が、別のエージェントであれば、エージェント間でコマンドをやり取りすることになります。xlsv は TCP/IP コネクションによってこのストリームを確立し、異なるマシン上のプロセス間でコマンドをやり取りするためのパッシブモードサーバです。

xlsv は、立ち上げ時にあるポートをパッシブオープンします。通常 LANDSCAPE サーバは、xlconf/std/agent.xml に指定されているポートをオープンします。

続いて、このポートにアクティブコネクションしてきたストリームに対し、状態を返答します。アクティブモード側のエージェントはその状態が許可 (allow) である場合、SetAgent コマンドを使い、新しいエージェントを立ち上げます。以降、アクティブモード側のエージェントはこの新しいエージェントと通信するようになります。その様子を 表 1.1 に示します。

(2) における xlsv の IP アドレス認証によって IP アドレスが不許可である場合、以下のメッセージを出し切断します。

表 1.1: XL プロトコルの起動

	アクティブ側エージェント		xlsv
(1)			xlsv 起動時に port をパッシブオープン
(2)	コネクションをアクティブオープン	->	IP アドレスを認証
(3)	改行を送信	->	XL プロトコルと判断
(4)		<-	サーバの状態をチェックし、状態を返答。
(5)	SetAgent により起動エージェントを指定	->	ユーザ認証を行いエージェントを起動

`<Permission>Denied</Permission>`

また、サーバが高負荷であった場合は、

`<Permission>Busy</Permission>`

と返答します。認証されると、(3)に進み、サーバ側は待ち状態に入ります。(4)においての返答は、

`<Permission>Allow</Permission>`

に固定されています。(5)における、SetAgent の動作については、リファレンスマニュアル (2.3.1 節) を参照してください。SetAgent 以降の XL コマンドは、立ち上げたエージェントの機能によります。各エージェントのマニュアルを参照してください。

1.5 xlsv の HTTP エージェント機能

一方、xlsv は HTTP プロトコルを受ける機能を持っています 表 1.2。

HTTP エージェント機能は、XL プロトコルと同じポートに対してパッシブオープンすることが出来ます。接続されてきたプロトコルが、XL か HTTP かを判断するのは、(3)において、最初の一文字が、改行か、メソッドの最初の一文字かで判断される。判断された以降は、HTTP のプロトコルとして動作する。

HTTP エージェント機能では、xlsv の立ち上げ時 (1) に、どの URL でリクエストされたパスに様々なエージェントを割り当てよう初期化することが出来る。そのときの関数が HttpAgent (2.3.2 節) です。

その上で xlsv は受け取った HTTP のヘッダおよび本体を、以下の一定の規則で、XL スクリプトに置き換えます。

1. HTTP ヘッダのメソッドをシンボルとする。
2. HTTP リクエスト時の URL を dir 属性に与える。
3. HTTP のバージョンは、http_ver という属性へセットされる。
4. HTTP のリビジョンは、http_rev という属性へセットされる。
5. Content-Type 以下、HTTP のヘッダ情報は、その名前と同じ属性へセットされる。

表 1.2: HTTP プロトコルの起動

	アクティブ側エージェント		xlsv		待機エージェント
(1)			xlsv 起動時に port をパッシュオープン	<->	xlsv により待機エージェントが立ち上げられ初期化される。
(2)	コネクションをアクティブオープン	->	IP アドレスを認証		
(3)	GET,POST などメソッドを送信	->	HTTP プロトコルと判断		
(4)	以降 HTTP ver.1.0./1.1.に従う		与えられたパス名から対応するエージェントを選択		
(5)			HTTP ヘッダ情報および本体情報を XL スクリプトに変換エージェントへ渡す。	->	
(5)	HTTP ドキュメントを受け取る。	<-	返答を HTTP の返答へ変換、必要あらばコネクション切断	<-	情報を処理し、xlsv へ返答を返す。

6. 情報本体は、全体を一つの XLT_RAW タイプデータとして引数に与える。(ver.B.b11 以降のみ)

先に (1) において起動されている対応エージェントは、URL のパスが割り付けられている。xlsv は、対応するパスのエージェントを探します。対応するエージェントがない場合は、ブラウザ側に 404 エラーを返します。対応するエージェントがある場合は、上記変換されたメソッドを対応するエージェントの HTTP (3.2.1 節) 環境において評価する、以下の評価式をエージェントに対して投げる。

```
<HTTP><method> body </method></HTML>
```

従って、エージェント側は、上記 HTTP のメソッドに対応した関数を環境 HTTP (3.2.1 節) に用意しておく必要がある。このエージェント側にあるべき関数に関しては、[UNDEF REF (xlsv-HTTP-function)]を参照していただきたい。エージェントはこれらのすべての関数を用意する必要は無く、この環境に用意されていないメソッドが呼ばれた場合は、503 エラーとしてブラウザに返答されます。

エージェントのメソッド関数の返すデータはまた一定の書式に乗っ取っていなければなりません。基本的に、マイムコード (text/html, image/gif など) をシンボルとした、XL データ形式です。詳細は、 4 節を参照してください。

第2章 xlsv 基本XL関数

2.1 概要

エージェントを起動する `SetAgent` と、HTTP エージェント機能のセットアップを行う `HttpAgent` が主な関数です。

2.2 継承情報

xlsv エージェントは standard エージェント [UNDEF REF (xl-agent)] の機能をそのまま継承している。

2.3 関数

2.3.1 SetAgent (エージェントの起動)

プロトタイプ

`<SetAgent> agent type </SetAgent>`

引数

agent [1] [文字列]

type [1] [文字列] user/root/server

属性

client [任意] クライアントのバージョン

評価形式

applicative

所属エージェント

xlsv

所属環境

Env1

説明

agent で与えられた名前のエージェントを起動します。どのようなエージェントが起動可能かは、2.3.3 節により定義しておく。LANDSCAPE SERVER においては、`xlconf/std/agent.definition.xl` にまとめてある。*type* は以下の通り d えます。

1. user

ユーザクライアントからの接続要求。必要があれば、ユーザ名、パスワードが求められる。

2. server

サーバからの接続要求。必要があれば、ユーザ名、パスワードが求められる。user か server かはクライアントの自己申告。

3. root

ルートモードでの接続要求、パスワードが求められる。

戻り値

エラー時、xlsv に `XLT_ERROR` が返る。成功時、対応するエージェントが xlsv 内（そのもの）である場合は、文字列 "ok" が返る。対応する外部のエージェントが立ち上がった場合は、そのエージェントから、`XLT_NULL` が返る。

エラー

- `XLE.SEMANTICS_TYPE_MISMATCH` "type mismatch result of GetUserInfo"

認証時のエラー

- `XLE.PROTO_PERMISSION_DENIED` "permission denied"

認証の結果、接続不許可。

- `XLE.PROTO_LAUNCH` "cannot launch"

なんらかシステムの問題でエージェントを立ち上げることが出来なかった。

参考

`xlconf/std/agent.definition.xl`

バグ

2.3.2 HttpAgent (HTTP エージェント機能の設定)

プロトタイプ

```
<HttpAgent> cmd.... </HttpAgent>
```

引数

cmd [0+] 初期化コマンドリスト

属性

remote [任意ただし条件によっては必須] 起動エージェントの指定

path [必須] 割り当てるパス

alias [任意] 条件の複製 (同一エージェントの指定)

評価形式

normal

所属エージェント

xlsv

所属環境

Env1

説明

remote で指定したエージェントを、HTTP の処理系として起動し、そのエージェントを、*path* で指定する URL パスに割り当てます。*cmd...* に指定する、初期化コマンドは、エージェントの立ち上げ時にエージェントに送り込まれるコマンドです。ここに設定可能なエージェントは XL エージェントの起動の方法で起動されるので、起動可能エージェントとして登録されているエージェントに限ります。さらに HTTP のリクエストとエージェントをつなぐ作法はあらかじめ決められているので、その作法を実装したエージェントのみが指定出来ます。

alias は、異なるパスで (*path*) 同一の機能を実現したいときに、同一のエージェントが割り当てられたパスを指定する。この場合、*remote* 属性は不要になる。同一のエージェントに対してことなる設定を送り込むのでそれらを切り分けられる仕組みが実装されているエージェントである必要がある。

remote の書式は URL でしめす。エージェントは、ディレクトリパス最初のエレメントに@を入れ、指定する。例えば、

```
xlp://localhost:8080/@gbview/
```

戻り値

正常終了は XLT_NULL。エラー時は以下に挙げるエラーが返る。

エラー

- XLE_PROTO_INV_PARAM "path attribute"

属性 *path* が指定されていません。

- XLE_PROTO_INV_PARAM "alias attribute"

alias に指定されたパスの設定が存在しなかった。

- XLE_PROTO_INV_PARAM "cannot initialize remote"

初期化出来なかった。

参考

xlconf/std/agent.definition.xl

バグ

2.3.3 DefineAgent (起動可能エージェントの設定)

プロトタイプ

```
<DefineAgent> group agent type environment function </DefineAgent>
```

引数

group [1] 文字列型

agent [1] 文字列型

type [1] 文字列型 user/server/root

environment [1] XL_NULL または、XL_ENV

function [1] XL_NULL または、XL_FUNC

属性

passwd [任意] passwd/anonymous default=passwd

dircheck [任意] on/off default=off

評価形式

applicative

所属エージェント

xlsv

所属環境

Env1

説明

agent は起動するエージェントのエージェント識別名を与える。必ずしも起動するプロセス名とは一致しなくても良い。

group エージェントのグループ名。現在使われていない機能。なにを指定しても良い。

type エージェントの起動可能タイプ、SetAgent 時に指定する *type* と一致する必要がある。

environment XL_NULL の場合は SetAgent により起動される時点での環境で、起動関数 *function* が評価される。XL_ENV が指定されている場合は、その環境で *function* が評価され、その結果が、プロセスのコマンドラインとして execp などで起動される。

function XL_NULL の場合、外部にエージェントは起動されず、xlsv 自身に *environment* で与えられた環境のインタプリタが、SetAgent を発行しているストリームに生成される。XL_FUNC が指定されている場合、*environment* で *function* が評価される。

戻り値

正常終了は XL_NULL

エラー

- XLE_SEMANTICS_TYPE_MISMATCH

引数の型が違う。

参考

xlconf/std/agent.definition.xl

バグ

第3章 HTTP エージェント呼び出し規定 (HTTP リクエスト関数)

3.1 概要

xlsv の HTTP エージェント機能が立ち上がっている場合、HTTP からのリクエストが XL 関数に変換され、エージェントで評価されます。そのときの関数呼び出し規定をここで示します。HTTP エージェントの初期化は、HttpAgent 関数で指定した関数そのまま送り込まれるので、各エージェントの要求する通り、HttpAgent の要素に諸機能関数を指定しておけばよいです。一方、HTTP リクエストに対して、エージェントに送り込まれる XL 関数は HTTP リクエストより形式的に変換された XL 関数であり、これらは、この規定に従って各エージェントに定義されていなければいけません。

この章に掲げられた関数群は、xlsv に定義されている物ではなく、xlsv の HTTP エージェント機能により呼び出されたエージェントに定義されているべき関数です。

3.2 関数

3.2.1 HTTP (HTTP リクエスト関数用環境)

プロトタイプ

<HTTP></HTTP>

引数

GET, POST, PUT, など

属性

環境呼び出し属性に従う。

評価形式

environment

所属エージェント

HTTP エージェント

所属環境

Env1

説明

xlsv の HTTP エージェントは、必ずこの環境を評価環境として、各リクエスト関数を呼び出します。HTTP エージェントはこの環境を Env1 内に定義し、さらにこの HTTP 環境内にリクエスト関数を定義しておく必要があります。

xlsv はこの環境を呼び出すときに、この環境の子、リクエスト関数の親となるテンポラリな環境に、シンボル `HttpAgent_path`, `HttpAgent_remote` を設定し、そこに、`HttpAgent` で指定した `path`, `remote` を設定する。これにより、どの設定が呼ばれているかがわかります。

戻り値

なし

エラー

なし

参考

バグ

3.2.2 GET

プロトタイプ

<GET/>

引数

属性

http_ver [必須] HTTP のバージョン

http_rev [必須] HTTP のリビジョン

HTTP リクエストヘッダメンバ [任意]

評価形式

applicative

所属エージェント

HTTP エージェント

所属環境

HTTP (3.2.1 節)

説明

HTTP からのリクエスト GET が *xlsv* により、当該関数に変換されます。各 HTTP エージェントにより環境 HTTP (3.2.1 節) 内に定義されていない関数です。

戻り値

この関数の機能自体は、HTTP エージェントごとに個別に決められるものです。それらの機能によって戻るデータは再び HTTP の返答へ変換されるため、[UNDEF REF (*xlsv*-result-code)] に従った戻り値である必要があります。エラーを返すときには、*XL_ERROR* をそのまま返せば、これを *xlsv* が解釈し、HTML 形式でブラウザ側に返答します。

エラー

HTTP エージェントの規定による。

参考

バグ

3.2.3 POST

プロトタイプ

<POST> raw </POST>

引数

raw [1+] XL_RAW

属性

http_ver [必須] HTTP のバージョン

http_rev [必須] HTTP のリビジョン

HTTP リクエストヘッダメンバ [任意]

評価形式

applicative

所属エージェント

HTTP エージェント

所属環境

HTTP (3.2.1 節)

説明

HTTP からのリクエスト POST が *xlsv* により、当該関数に変換されます。各 HTTP エージェントにより環境 HTTP (3.2.1 節) 内に定義されていない関数です。

raw は、POST の HTTP 本体のデータがそのまま入った XL_RAW 型変数です。

戻り値

この関数の機能自体は、HTTP エージェントごとに個別に決められるものです。それらの機能によって戻るデータは再び HTTP の返答へ変換されるため、[UNDEF REF (*xlsv*-result-code)] に従った戻り値である必要があります。エラーを返すときには、XL_ERROR をそのまま返せば、これを *xlsv* が解釈し、HTML 形式でブラウザ側に返答します。

エラー

HTTP エージェントの規定による。

参考

バグ

3.2.4 PUT

プロトタイプ

<POST> raw </POST>

引数

raw [1+] XL_RAW

属性

http_ver [必須] HTTP のバージョン

http_rev [必須] HTTP のリビジョン

HTTP リクエストヘッダメンバ [任意]

評価形式

applicative

所属エージェント

HTTP エージェント

所属環境

HTTP (3.2.1 節)

説明

HTTP からのリクエスト POST が *xlsv* により、当該関数に変換されます。各 HTTP エージェントにより環境 HTTP (3.2.1 節) 内に定義されていない関数です。

raw は、POST の HTTP 本体のデータがそのまま入った XL_RAW 型変数です。

戻り値

この関数の機能自体は、HTTP エージェントごとに個別に決められるものです。それらの機能によって戻るデータは再び HTTP の返答へ変換されるため、4 節に従った戻り値である必要があります。エラーを返すときには、XL_ERROR をそのまま返せば、これを *xlsv* が解釈し、HTML 形式でブラウザ側に返答します。

エラー

HTTP エージェントの規定による。

参考

バグ

第4章 HTTP リクエスト関数の戻り値

4.1 概要

HTTP リクエスト関数の戻り値は基本的に MIME CODE のシンボルを持ったリストである。

4.2 XML 要素

4.2.1 text/html

プロトタイプ

(text/html *data*)

内部要素

data [1] XL_STRING / リスト

属性

なし

所属エージェント

xslv HTTP エージェント

要素パス表現

/text/html

説明

data が HTML であることを示します。 *data* が XL_STRING である場合、xslv は *data* そのものをブラウザ側に返答します。 *data* がリストである場合、xslv は *data* が HTML 型の XL 表現であると解釈し、HTML に変換してブラウザ側に送信します。

参考

バグ

4.2.2 text/xml

プロトタイプ

(text/xml *data*)

内部要素

data [1] XL_STRING / リスト

属性

なし

所属エージェント

xslv HTTP エージェント

要素パス表現

/text/xml

説明

data が XML であることを示します。 *data* が XL_STRING である場合、xslv は *data* そのものをブラウザ側に返答します。 *data* がリストである場合、xslv は *data* が XML 型の XL 表現であると解釈し、HTML に変換してブラウザ側に送信します。そのとき、*data* は必ず以下の形をしていると解釈します。

```
(  
  <?xml version="..." encoding="..."?>  
  <root-tag>  
    .....  
  </root-tag>  
)
```

ただし、root-tag は何らかの規定によって定められる、XML の root-tag です。このとき、タグ?xml の encoding 属性を xslv はチェックし、そのコード系に合わせてヘッダなどを出力します。

参考

バグ

4.2.3 image/jpeg

プロトタイプ

(image/jpeg *data*)

内部要素

data [1] XL_RAW

属性

なし

所属エージェント

xslv HTTP エージェント

要素パス表現

/image/jpeg

説明

data が JPEG データであることを示します。xslv は *data* の内容をバイナリデータとしてブラウザ側に返答します。

参考

バグ

4.2.4 image/gif

プロトタイプ

(image/gif *data*)

内部要素

data [1] XL_RAW

属性

なし

所属エージェント

xlsv HTTP エージェント

要素パス表現

/image/gif

説明

data が GIF データであることを示します。xlsv は *data* の内容をバイナリデータとしてブラウザ側に返答します。

参考

バグ

関連図書

- [1] 森洋久. COSMOS スタートアップ・マニュアル. GLOBALBASE PROJECT, 2006.
- [2] 森洋久. LANDSCAPE スタートアップ・マニュアル. GLOBALBASE PROJECT, 2006.

履歴

1. 日時: 2006-07-26
マニュアル生成。(2006-07-26 版)
2. 日時: 2006-07-25
著者: 森 洋久 反映されたバージョン: ver.B.b11
このマニュアルを作成