Protocols, communication format, and PHP script specification for Palm/PHProjekt synchronization software

Revision: 1.9

Date: 2003/11/18 13:34:46

Contents

1 Introduction 4
2 Scope 4
3 Concept of operation 4
4 Location of the PHP scripts 5
5 Format specifications 5
5.1 General format specifications 5
5.2 Format for usercal.php 6
5.3 Format for usercal-delete.php 8
5.4 Format for usercal-modify.php 8
5.5 Format for usercal-new.php 9
6 Success and error status 9
6.1 Success status 9
6.2 Error status 10

1 Introduction

The purpose of this document is to specify the format and protocols for the transfer of data between PHP scripts accessing a PHProjekt database and Palm conduit. The purpose of the transfer is to synchronize the PHProjekt database and the Palm database.

2 Scope

This document describes the format of the data transferred, and the protocols for transferring data. This document also describes the format and protocols for the PHP scripts to signal the status of a transfer to the conduit. The status can be success, or an error condition.

3 Concept of operation

There are three PHP scripts, each of which performs a single function:

1. usercal.php

   This script returns the contents of every record for future events in the PHProjekt database for a given user.

2. usercal-delete.php

   This script removes a single record from the PHProjekt database, and returns a status to the conduit.

3. usercal-modify.php

   This script modifies an existing record in the PHProjekt database, and returns a status to the conduit.

4. usercal-new.php

   This script creates a new record in the PHProjekt database, and returns a status to the conduit.

To synchronize the PHProjekt database with the Palm database, the Palm user executes the conduit application (conduit). The conduit invokes the PHP scripts by accessing the URL for that script. The conduit accesses

the PHProjekt database by invoking the PHP scripts. The conduit obtains the most recent list of records by invoking usercal.php, and then invokes the other scripts to create, modify, or delete records on the PHProjekt database as necessary for synchronization.

4 Location of the PHP scripts

The PHP scripts are located in a subdirectory of the PHProjekt directory. The name of the subdirectory is "sync".

If the URL for PHProjekt is

```
http://www.shcorp.com/phprojekt
```

then the URL for the usercal.php script would be

```
http://www.shcorp.com/phprojekt/sync/usercal.php
```

5 Format specifications

5.1 General format specifications

The encoding for URLs is specified in RFC 1738 (Uniform Resource Locators). Characters that are reserved or unsafe and the space character) should be encoded. These characters are:

```
; / ? : @ = & < > # { } | ^ ~ [ ] % \
```

and space. A character is encoded using three bytes: %xx where xx are the hexadecimal digits representing the ASCII code of the character. Therefore, to pass a user name of `test&name` and user password of `test%pw` to usercal.php, the following URL would be used:

```
http://www.shcorp.com/usercal.php?userName=
test%26name&userPW=test%25pw
```

The format for a date specification is

```
yyyy:mm:dd
```

where mm has the range 1-12, and dd has the range 1-31. The range of dd depends on the month mm. The database will return an error if the date is illegal (such as 2001:02:31).

The format for a time specification is

```
hh:mm
```

The time is based on a 24 hour clock.

5.2 Format for usercal.php

This script is invoked as

```
usercal.php?userName=NAME&userPW=PASSWD&reqTime=TIME&
reqDate=DATE
```

where NAME is a string specifying the user's name in the PHProjekt database, PASSWD is the user's PHProjekt password, TIME is a time specification, and DATE is a date specification.

The script first checks that user NAME exists, and that PASSWD is correct. Next, the script checks that the DATE and TIME specifications are legal. An error status (section 6.2) is returned in case any of the checks fails.

If there is no error, the script returns the values of all the records in the table termine where:

1. The user name is NAME, and
2. The end field of the record is later than the present time.

The present time and date is either the time and date specified by TIME and DATE, or the system time, whichever is earlier.

The format of the values is as follows:

```
<events>
  <event>
    <id>Event1.ID</id>
    <date>Event1.Date</date>
    <begin>Event1.Begin</begin>
    <end>Event1.End</end>
    <desc>Event1.Desc</desc>
```

```
    <note>Event1.Note</note>
  </event>
  <event>
    <id>Event2.ID</id>
    <date>Event2.Date</date>
    <begin>Event2.Begin</begin>
    <end>Event2.End</end>
    <desc>Event2.Desc</desc>
    <note>Event2.Note</note>
  </event>
   .
   .
   .
  <event>
    <id>EventN.ID</id>
    <date>EventN.Date</date>
    <begin>EventN.Begin</begin>
    <end>EventN.End</end>
    <desc>EventN.Desc</desc>
    <note>EventN.Note</note>
  </event>
</events>
```

- `Event`X`.ID` is the PHProjekt record ID for event X.
- `Event`X`.Date` is the date specification for the date for event X.
- `Event`X`.Begin` is the time specification for the start time for event X.
- `Event`X`.End` is the time specification for the end time for event X.
- `Event`X`.Desc` is the description for event X.
- `Event`X`.Note` is the note for event X.

If there are no records to return, the script returns

```
<events>
</events>
```

5.3 Format for usercal-delete.php

If there is no begin time, then the script returns

```
<begin>----</begin>
```

as the specification of the beginning time of the record.
This script is invoked as

```
usercal-delete.php?userName=NAME&userPW=PASSWD&
recID=ID
```

where `NAME` is a string specifying the user's name in the PHProjekt database, `PASSWD` is the user's PHProjekt password, and `ID` is the PHProjekt record ID of the record to delete.

The script first checks that user `NAME` exists, and that `PASSWD` is correct. It then attempts to delete the specified record, and checks that the record was successfully deleted. An error status (section 6.2) is returned in case any of the checks fails.

If there is no error, the script returns a status indicating success (section 6.2).

5.4 Format for usercal-modify.php

This script is invoked as (as a single line)

```
usercal-modify.php?userName=NAME&userPW=PASSWD&
recID=ID&recDesc=DESCRIPTION&recNote=NOTE&
recDate=DATE&recBegin=BEGIN&recEnd=END
```

where `NAME` is a string specifying the user's name in the PHProjekt database, `PASSWD` is the user's PHProjekt password, and `ID` is the PHProjekt record ID of the record to delete. `NAME`, `PASSWD`, and `ID` are required. The optional parameters are `DESCRIPTION`, which is the new value for the Description field, `NOTE`, which is the new value for the Note field, `DATE`, which is the date specification for the Date field, `BEGIN`, which is the time specification for the Begin field, and `END`, which is the time specification for the End field. If any of the optional parameters are present, the script updates the associated field of the specified record. If an optional parameter is not present, the script does not update the associated field of the specified record.

The script first checks that user `NAME` exists, and that `PASSWD` is correct. It then checks that the date and time specifications have the correct format, and that the record is not read-only. Finally, the script modifies the record and checks that the record was successfully modified. An error status (section 6.2) is returned in case any of these checks fails.

If there is no error, the script returns a status indicating success (section 6.2).

5.5 Format for usercal-new.php

This script is invoked as

```
usercal-new.php?userName=NAME&userPW=PASSWD&
recDesc=DESCRIPTION&recNote=NOTE&recDate=DATE&
recBegin=BEGIN&recEnd=END
```

where `NAME` is a string specifying the user's name in the PHProjekt database, `PASSWD` is the user's PHProjekt password, `DESCRIPTION` is the value for the Description field, `NOTE` is the value for the Note field, `DATE` is the date specification for the Date field, `BEGIN` is the time specification for the Begin field, and `END` is the time specification for the End field. The required parameters are `NAME`, `DATE`, `BEGIN`, `END`, and one of `NOTE`, or `DESCRIPTION`. If an optional parameter is not present, the associated field of the newly-created record will have a NULL value.

The script first checks that user `NAME` exists, and that `PASSWD` is correct. It then checks that the date and time specifications have the correct format, and finally that the record was successfully created. An error status (section 6.2) is returned in case any of these checks fails.

If there is no error, the script returns a status indicating success and the PHProjekt record ID number (section 6.2).

6 Success and error status

6.1 Success status

If a `usercal.php` executes successfully, it returns a (possibly empty) list of records.

If `usercal-new.php` executes successfully, it returns

```
SUCCESS: ID
```

where `ID` is the PHProjekt record ID of the newly-created record.

The remaining scripts return

```
SUCCESS
```

upon successful execution.

6.2 Error status

All error status returns consist of a single line of ASCII text.

If the user `NAME` does not exist, then

```
DATA_ERROR: User NAME does not exist
```

is returned. If the password `PASSWD` is incorrect, then

```
DATA_ERROR: Illegal password
```

is returned. If the date or time specification is illegal, then

```
PROGRAM_ERROR: Illegal date specification:  'DATE'
```

or

```
PROGRAM_ERROR: Illegal time specification:  'TIME'
```

is returned. If the PHProjekt record with the specified user and record ID does not exist, then

```
DATA_ERROR: There is no record 'ID' with user 'NAME'
```

is returned. If there is an error creating, modifying, or deleting a PHProjekt record, then

```
DATA_ERROR: MESSAGE
```

is returned. `MESSAGE` is a text message describing the error.