

Site Controller

PDK

Date: December 2, 2020



Verifone®

Verifone Confidential

Verifone, Inc.
88 West Plumeria Drive
San Jose, CA 95134
Telephone: 408-232-7800
<http://www.verifone.com>

© 2020 Verifone, Inc. All rights reserved.

No part of this publication covered by the copyrights hereon may be reproduced or copied in any form or by any means - graphic, electronic, or mechanical, including photocopying, taping, or information storage and retrieval systems - without written permission of the publisher.

The content of this document is subject to change without notice. The information contained herein does not represent a commitment on the part of Verifone. All features and specifications are subject to change without notice.

Verifone, Ruby SuperSystem, and Secure PumpPAY are registered trademarks of Verifone, Inc. Ruby Card, iOrder, and Commander Site Controller are trademarks of Verifone. All other brand names and trademarks mentioned in this document are the properties of their respective holders.

Revision History

Date	Description
09/06/2017	Converted document from Sapphire to Site Controller to include Commander.
12/02/2020	Glossary of Terms - Changed Sapphire definition to mention it is legacy and EOL.

Introduction to PDK-Site Controller

The Partner Development Kit for Site Controller (PDK-Site Controller) provides a set of scripts for developers at third-party back office software companies who develop and use applications that interface with Verifone, Inc. Site Controller systems. The documentation provided with PDK-Site Controller includes the following:

- *Site Controller Windows Scripting Host Reference* — Contains four separate Windows[®] Scripting Host scripts.
- *Site Controller HTTP Primer* — An introduction to the Hyper-Text Transfer Protocol (HTTP) that explains how to use it from Java, JavaScript, or VBScript to write programs for the Site Controller application.
- *Site Controller URL Reference* — Detailed information about using HTTP for performing Site Controller maintenance and reporting functions.
- *Site Controller Style Sheets Reference Manual* — An introduction to the XML style sheets (XSLT) used with the Site Controller application.
- *Site Controller Gap Analysis* — A reference that identifies all of the potential data items and the mechanisms that support them.

Documentation Conventions

The PDK-Site Controller manuals use the following documentation conventions:

Convention	Explanation / Example
http code	Type the code as written. For readability, URLs may be presented on two lines, but should be entered as one continuous string. https://{site-controller_IP}/cgi-bin/CGILink?cmd=validate&user=manager&passwd=123
{cookie}	Replace curly brackets (braces) and text within the curly brackets with a value.

Glossary of Terms

The following terms and definitions will assist the reader with understanding the content of the Site Controller URL Reference.

Glossary of Terms		
Term	URL Syntax	Definition
Site Controller	{site-controller_IP}	A generic term for Sapphire and Commander and applies to both.
Commander	{commander_IP}	It is the new generation of the Verifone Site Controller. The Commander specifically operates with the Ruby2 and the Topaz POS workstations.
Sapphire	n/a	Sapphire is the legacy generation of the Verifone Site Controller. Sapphire is an End-of-Life product as of June 25, 2020. References to Sapphire in the URL Reference are due to the use of "sapphire" as namespace definitions, or for identifying deprecated items.

Accessing the Site Controller

The reporting and maintenance functions of the Site Controller are accomplished by exchanging XML documents with the Site Controller using the Hyper-Text Transfer Protocol (HTTP). Use GET requests to retrieve data from the Site Controller and POST requests to update datasets on the Site Controller.

The initial request in a Site Controller document exchange session is always for a credential document. After all the work in a session has been completed the credential must be released.

Getting Credentials

Requesting a credential document is the equivalent of a log in and requires a username and password.

The retrieval of the credential is vital because it contains a “cookie” (an ASCII hexadecimal value) that is required for all subsequent GET and POST requests. This credential request can be made directly from the Web browser by using the following syntax:

`https://{site-controller_IP}/cgi-bin/CGILink?cmd=validate&user={user}&passwd={passwd}`

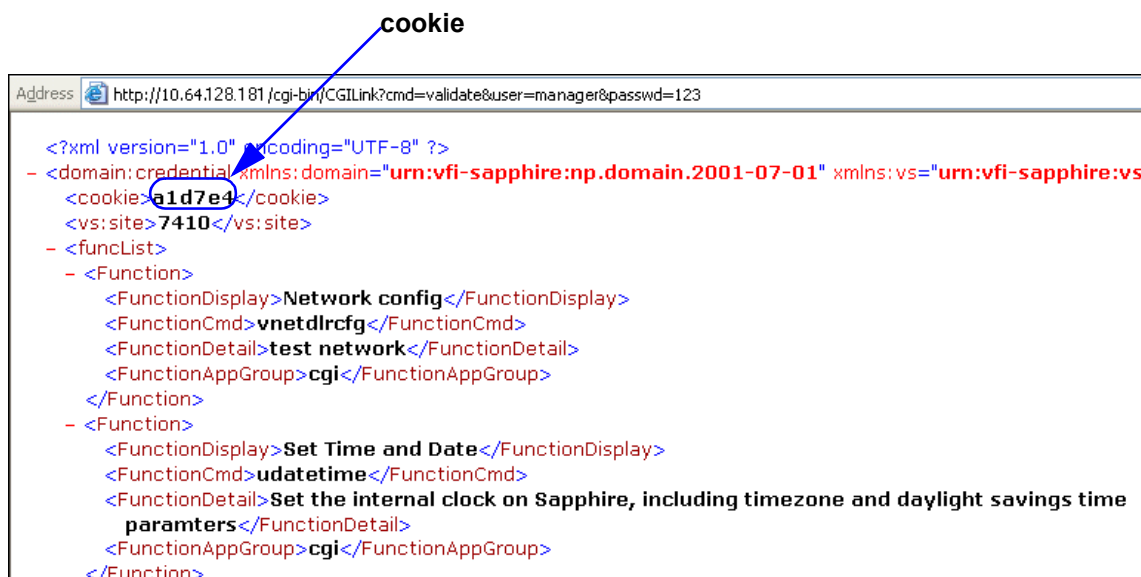
Example:

1. Enter the following in the address bar of the browser:

```
https://10.64.128.181/cgi-bin/CGILink?  
cmd=validate&user=manager&passwd=123
```

Note: *The examples in the PDK-Site Controller documents use ‘10.64.128.181’ as the IP address of a targeted Site Controller, and ‘a1d7e4’ as the cookie. These values are for illustration purposes and must be replaced with appropriate values.*

2. Click **GO**. A credential document appears.



Note: For information about the credential document, see “Credential Management Functions” in the Site Controller URL Reference.

In addition to the cookie, the credential document also contains a list of actions available to the user who requested the credential (logged in). That user can access only the GET and POST commands listed in the credential. Site Controller returns an error document for any attempt to access a URL that is not available in the cookie’s function list.

Releasing Credentials

The final request for a Site Controller session must be to release the credential. After all the work in a document exchange session has been completed, the credential *must* be released.

To release a credential, enter this command using the following syntax:

`https://{site-controller_IP}/cgi-bin/CGILink?cmd=releaseCredential&cookie={cookie}`

Example:

`https://10.64.128.181/cgi-bin/CGILink?cmd=releaseCredential&cookie=a1d7e4`

Note: The examples in the PDK-Site Controller documents use ‘10.64.128.181’ as the IP address of a targeted Site Controller, and ‘a1d7e4’ as the cookie. These values are for illustration purposes and must be replaced with appropriate values.

Using GET and POST Commands

Access the Site Controller by sending HTTP GET and POST requests to a Common Gateway Interface (CGI) application. Site Controller uses four CGI applications:

- **CGILink** — Accepts only GET requests. CGILink is used to retrieve (view) datasets (*with the exception of the PLU dataset*) from the Site Controller and to obtain a credential document.
- **CGIUplink** — Accepts only POST requests. CGIUplink is used to post (update) datasets (*with the exception of the PLU data set*) to Site Controller. Reports and transactions cannot be posted to Site Controller through CGI, only datasets.
- **CGIPLULink** — Accepts only POST requests. CGIPLULink is used to retrieve (view) and post (update) PLU related data. The retrieval of PLUs requires a post, due to the fact that a “query” document is sent to Site Controller to provide the ability to retrieve a specific set of PLUs.
- **NAXML** — Accepts GET and POST. NAXML is used to retrieve (view) and post (update) documents conforming to the Conexus POS-BO standards. The schemas are not part of the PDK and must be retrieved from Conexus.

GET Commands

GET commands are used to view data from a Site Controller and to obtain a credential document.

Most browser commands are simple GET requests. All GET requests can be issued from the browser address bar directly. The browser turns a URL into a sequence of lines that make up an HTTP GET command. The URL should be valid and separate from the sequence of lines.

The syntax for a GET request is:

```
https://{site-controller_IP}/cgi-bin/CGILink?  
cmd={command}&cookie={cookie}
```

Example:

```
https://10.64.128.181/cgi-bin/CGILink?  
cmd=vpayrollpdlist&cookie=a1d7e4
```

Note: Site Controller GET command names begin with a ‘v.’

POST Commands

POST commands are used to update datasets and to retrieve and update PLU-related data. When a POST command is used, a payload must be supplied; that is, the XML document that contains the data to be updated on the Site Controller.

The following is an example of a POST command:

`https://10.64.128.181/cgi-bin/CGIUplink`

Followed by the start of the payload:

```
cmd=ufuelsite&cookie=a1d7e4
LF (linefeed)
LF
<?xml version="1.0" encoding="UTF-8"?>
<document>
.
. <content/>
.
</document>
```

***Note:** Site Controller POST command names begin with a ‘u.’*

Requesting Reports

Up to this point only datasets have been mentioned, but reports can also be requested. Most reports are derived from a database and returned as XML documents, with a few reports coming from the transactionSet/transSet (T-Log or electronic journal). Unlike retrieving datasets, retrieving reports requires additional parameters as well as a request for an additional document that contains a list of reports available.

Requesting Database Reports

Database reports (also known as Ruby Reports) are available for up to four periods. Each lower period is “rolled up” into the next higher. Period 1 is the lowest, and period 4 is the highest. A period 2 report contains totals for all period 1 reports closed since the last period 2 close and so on.

Prior to requesting database reports, you can request a list of available reports by issuing the **vreportpdlist** command using the following syntax:

```
https://{site-controller_IP}/cgi-bin/CGILink?
cmd=vreportpdlist&cookie={cookie}.
```

The document returned contains a list of all reports available with a set of parameters that you need when requesting a given report. All closed reports are named in a `yyyy-mm-dd.nnn` format, where `nnn` is the sequence number of the report.

Note: *You do not need to request a list of reports if you already know the report number. You can use the **reptnum** argument to request a specific report.*

The following example uses the information from the credential and **vreportpdlist** XML documents to request a closed period 1 department report:

```
https://10.64.128.181/cgi-bin/CGILink?  
cmd=vrubyrept&reptname=department&period=1&filename=2003-12-05.001&cookie=a1d7e4
```

The **cmd**, **reptname**, and **cookie** parameters came from the credential, while the **period** and **filename** came from the **vreportpdlist**.

Requesting Transaction Sets

A transaction set (transactionSet/transSet) is a log (electronic journal) of all transactions and journal events from the POS. Transaction sets are available only for period 1 and period 2, where a period 2 transaction set contains all the period 1 transactions since the last period 2 close. Prior to requesting the transaction set, a list of available reports can be requested by issuing the **vtlogpdlist** command:

```
https://{site-controller_IP}/cgi-bin/CGILink?  
cmd=vtlogpdlist&cookie={cookie}.
```

Note: *You do not need to request a list of reports if you already know the report number. You can use the **reptnum** argument to request a specific report.*

The document returned contains a list of all transaction sets available and a set of parameters that you need to use to request a given transaction set. All closed transaction sets contain a name in a `yyyy-mm-dd.nnn` format, where `nnn` is the sequence number of the report.

The following example uses the information from the credential and **vtlogpdlist** XML documents to request a closed period 1 transaction set:

```
https://{sapphire_IP}/cgi-bin/CGILink?  
cmd=vperiodrept&period=1&filename=2003-12-05.001&cookie=a1d7e4  
  
https://{commander_IP}/cgi-bin/CGILink?  
cmd=vtransset&period=1&filename=2003-12-05.001&cookie=a1d7e4
```

The **cmd** and **cookie** parameters come from the credential, while the **period** and **filename** come from the **vtlogpdlist**. The above request returns the **transactionSet/transSet** as an XML document. Since the transaction set can be rather large, you can also retrieve it in a *gzip* format by changing the cmd from **vperiodrept** to **vperiodreptz**, or **vtransset** to **vtranssetz**.

The raw transaction set can be difficult to work with. You can convert the transaction set to an easier to use T-Log format using the tlogDoc.xslt XML style sheet. The path to the tlogDoc.xslt style sheet is:

https://{site-controller_IP}/templates/vfit/transactionSet/tlogDoc.xslt

Note: *See the Site Controller Style Sheets Reference Manual for information.*

The transformation process must be conducted on the client; that is, when a **transactionSet/transSet** is requested, the Site Controller does not transform it to the T-Log format. Instead, the client side must transform the **transactionSet/transSet** document.