

Bring Your Own Projects for Polaris Systems!

Derek Brown (Rochester Hills Public Library)

Joe Fee (Westmoreland Library Network)



Introduction

- Derek Brown – Director of IT for the Rochester Hills Public Library
 - ◆ Single Branch Library with 2 Bookmobiles serving a population of 100,500
 - ◆ Circulation of around 2 million items
 - ◆ IT staff of 2 FTE and 2 PTE (College Students)
- Joe Fee – Technology Coordinator for the Westmoreland Library Network
 - ◆ 22 Libraries with 3 Branches and 1 Bookmobile serving a pop. of 355,000
 - ◆ Circulation of around 1.2 million items
 - ◆ IT staff of 2 FTE
- We wanted a platform to share projects openly with other Polaris users.
- Topics: Local School Database Sync to Polaris, Self-Built Polaris Telephony System, Wireless Access Manager Integration with Polaris, and more...



Rochester Hills Public Library and Rochester Public School Data Merge

- The objective of this project is to establish secured automated data exchanges of Rochester Community Schools Students' data between Rochester Community Schools (RCS) and Rochester Hills Public Library (RHPL) with a purpose of automatic students' enrollment into RHPL.
- Having an RHPL card provides students access to RHPL resources and allows the library and public schools to share services.
- Automating this process made it easier on RCS staff, teachers, parents, and librarians.



Functional Requirements and Technical Details

Solution Overview:

Data Merge Project	Description
Outbound Interface	Data feed from RHPL to RCS
Inbound Interface	Data feed from RCS to RHPL

Inbound Interface – file structure (inbound.txt) from RCS to RHPL

Field	Description	Data Type
RCSID	RSC Student ID	Int???
ActionCd	Action Code	CHAR (1)
FirstName	First Name	CHAR (30)
MiddleName	Middle Name	CHAR (30)
LastName	Last Name	CHAR (30)
Suffix	Suffix	CHAR (6)
Address	Mailing Address	CHAR (40)
City	City	CHAR (30)
Zip	Zip Code	CHAR (5)
BirthDt	Birth Date	DATE (8) (mmddyyyy)
HomePhoneNum	Home Phone Number	CHAR (10)

Outbound Interface – file structure (outbound.txt) from RHPL to RCS

Field	Description	Data Type
RCSID	RSC Student ID	Int???
RHPLCardNum	RHPL Card Number	CHAR (14)



Data Guidelines – Inbound

Field	Field Name	Data Type	Len	Required	Comments/Description
1.	RCSID	Int		Y	Universal Student ID. RCS will send a value
2.	ACTION CODE	Char	1	Y	RCS will send a value A – for all active students I – for inactive students
3.	FIRST NAME	Char	30	Y	RCS will send a value
4.	MIDDLE NAME	Char	30	N	If no value, field is included but left blank in the input file.
5.	LAST NAME	Char	40	Y	RCS will send a value
6.	NAME SUFFIX	Char	6	N	If no value, field is included but left blank in the input file.
7.	ADDRESS	Char	40	Y	RCS will send a value
8.	CITY	Char	30	Y	RCS will send a value
9.	ZIP	Nbr	5	Y	4 digit extender is not required.
10.	BIRTHDATE	Date	8	Y	RCS will send a value
11.	HOME PHONE NUMBER	Char	10	N	RCS may send a value Example: 2489999999

Data Guidelines – Outbound

Field	Field Name	Data Type	Len	Required	Comments/Description
1.	RCSID	Int		Y	Universal Student ID. RHPL will send a value
2.	RHPL Card Number	Char	14	Y	RHPL will send a value



Import Process Details

#	Task	Object	Schedule	What is done
1	Retrieve new RCS file from RCS FTP	Scheduled Task/WS Script - using WinSCP. Files are copied to G:\RCSDataFiles	Friday @ 9:28 pm	Scheduled Windows Task runs on Friday night and downloads new file from RCS to a local folder. Also moves server file to an Archive folder on RCS ftp server
2	Import RCS file to SQL	Scheduled SQL Server dstx job	Saturday @ 6:00 am	Process new RHPL.txt file and exports records to dbo.RHPLc temp table
3	Assign RHPL IDs to students without one	Scheduled SQL Server dstx job SQL Cursor code	Saturday @ 6:00 am	Looking at RHPLc table and assigns an available RHPL id from RHPLIDPool table to students who have blank RHPLid
4	Generate Import to Polaris file	Scheduled SQL Server dstx job calls RCSNewPolarisImport View that is created to match required syntax of Polaris offline registration file	Saturday @ 6:00 am	Generates polaris import file so new users can be imported to Polaris DB via Polaris Offline Registration manual process
5	RenameRCSFiles	VBScript/Scheduled Task	Saturday @ 7:20 am	Renames 3 files by adding date stamps to the names: <ul style="list-style-type: none"> - rhpl.txt→ rhplyyyymmdd.txt - Rochester.csv--> rochesteryyyymmdd.txt - RCSPolarisImport.trn→ RCSPolarisImportyyyyymmdd.trn

Import Process Details (Cont.)

6	RCSFtpAndMove	VBScript/WS Script	Saturday @ 8:30	Sends new RHPL file to RCS, moves RCSPolarisImportyyyymmdd.trn to C:\ProgramData\Polaris\5.0\OfflineTransaction and also archives RCS and RHPL files to G:\RCSDataFiles\Archive
7	Run Polaris Import Job	MANUAL		Requires manual support to import generated file to Polaris Offline Registration tool for processing. Follow Steps below.

Step 7 - Running Polaris Import File to generate new RCS accounts.

- 1) Log in to Polaris on Server
- 2) Choose Circulation → Bookmobile
- 3) Go to Tools → Options and choose option for All offline transaction/log files - click ok
- 4) Select RCSPolarisImportyyyymmdd.trn file under transactions to be uploaded window
- 5) Click on Upload and ignore errors recording them to the log



Task #1: Import Files

Task Scheduler Job Properties - RCSFtpscript - Notepad

File Edit Format View Help

```
# Automatically abort script on errors
option batch abort
# Disable overwrite confirmations that conflict with the previous
option confirm off
# Connect using a password
open
# Change the Local Directory
lcd "G:\RCSDataFiles"
# Change remote directory
cd /
# List Remote Files AA
ls
# Force binary mode transfer
option transfer binary
# Upload the files to current working directory
get Rochester*.*
mv /Rochester*.* /Archive/toRHPL/
# Disconnect
close
# Exit WinSCP
```

General Triggers Actions Conditions Settings History

When you create a task, you must specify the action that task property pages using the Properties command.

Action	Details
Start a program	C:\Scripts\rcsftptestscript.bat

rcsftptestscript - Notepad

File Edit Format View Help

```
@echo off
> c:\temp\RCS_File_Download-%DATE:~-4-%DATE:~4,2-%DATE:~7,2%_time:~0,2%%time:~3,2%%time:~6,2%.log (
cd "C:\Program Files (x86)\WinSCP\"
c:
WinSCP.com /script="C:\scripts\RCSFtpscript.txt" /log="C:\winscp.log
EXIT
)
@exit
```

Task #5: Rename Files

Action	Details
Start a program	G:\RCSPackages\renameRHPLtoRCSFile.vbs

renameRHPLtoRCSFile - Notepad

File Edit Format View Help

```
'=== Script to rename RHPL file to RCS with a datestamp ===
Dim strNewFileName, objFSO, strbibNewFileName
Set objFSO = CreateObject("Scripting.FileSystemObject")
'Get the folder where feed files are stored
Dim objFolder
Set objFolder = objFSO.GetFolder("G:\RCSDataFiles") ' <-- change this to the directory
Set objFSO = CreateObject( "Scripting.FileSystemObject" )
```

Task #2-4: Run SQL Steps

Job Properties - RCSPolarisImport

Select a page: General Steps Schedules Alerts Notifications Targets

Script Help

Job step list:

St...	Name	Type	On Success	On Failure
1	RCSstoRHPLImport	SQL Serv...	Go to the...	Quit the job...
2	AssignRHPLIDCursor	Transact...	Go to the...	Quit the job...
3	Demote non-RCS Students	Transact...	Go to the...	Quit the job...
4	RCSPolarisImportFileCreate	SQL Serv...	Go to the...	Quit the job...
5	RHPLtoRCSFile	SQL Serv...	Quit the j...	Quit the job...

Task #6: Upload Completed Files

Action	Details
Start a program	G:\RCSPackages\RHPLtoRCSFTPMove.bat

RHPLtoRCSFTPMove - Notepad

File Edit Format View Help

```
@echo off
> c:\temp\RCS_FTP_and_Move-%DATE:~-4-%DATE:~4,2-%DATE:~7,2%_time:~0,2%%time:~3,2%%time:~6,2%.log (
cd "C:\Program Files (x86)\WinSCP\"
c:
WinSCP.com /script="C:\scripts\RHPLtoRCSFtpscript.txt" /log="C:\winscp.log
if %ERRORLEVEL% neq 0 goto error
echo Upload succeeded, moving local files
move G:\RCSDataFiles\RHPL*.* G:\RCSDataFiles\Archive
move G:\RCSDataFiles\Rochester*.* G:\RCSDataFiles\Archive
move G:\RCSDataFiles\RCSPolarisImport*.trn C:\ProgramData\Polaris\5.6\OfflineTransaction
)
@exit
```



Downloads for RCS/RHPL Project

- <https://www.rhpl.org/IUG>
- This directory contains all documentation, processing files (DTSX), and other necessary pieces of code to mirror our project.
- Please feel free to contact with any questions:

Derek Brown (Director of IT @ Rochester Hills Public Library)

(248) 650-7123

Derek.Brown@rhpl.org



WCFLS Telephony

- The objective of this project is to:
 - ◆ reduce the cost of mailing overdue notices
 - ◆ reduce staff time
 - ◆ add new reminder notice for patrons without email addresses
 - ◆ add new hold pickup notice for patrons who still prefer mailing address
- Telephony Service saved over \$7,000 in postage and supplies



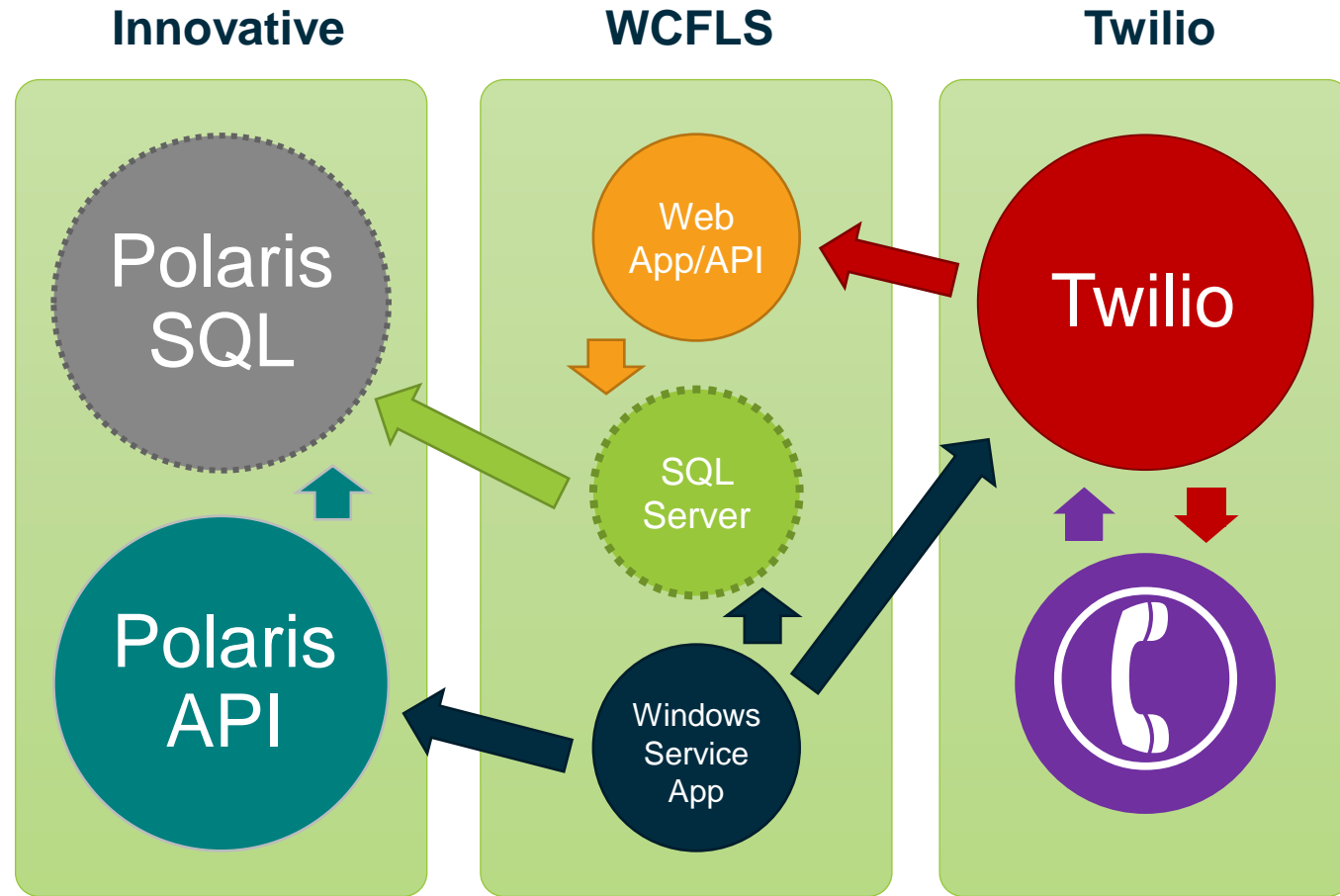
WCFLS Telephony Development

- Polaris SQL Server
- WCFLS SQL Server
- Windows Service Application
 - ◆ Developed using Visual Studios in Visual Basic.net
- Twilio Cloud Services
- Windows IIS Server
 - ◆ Web App/API – developed using ASP.net in Visual Basic
- Polaris API



WCFLS Telephony Process

1. Extract/Format Data (Polaris SQL to W-SQL)
 - a. Convert Print Holds to Telephone
 - b. Retrieve rows from Polaris SQL
 - c. Create reminders for non-email patrons
 - d. Create queue, group by patrons in W-SQL
2. Send call list to Twilio using Windows Service application
3. Twilio initiates call
 - a. Retrieves notification message from W-API if call is answered
 - b. Call updates to W-API
4. Update Polaris API with notification status from Windows Service



Polaris Data Outbound

- Query joining Results..NotificationQueue, Polaris..Patrons, and Polaris..PatronRegistration

Field	Description	Data Type
ItemRecordID	Unique ID for Item Record	INT
NotificationTypeID	Notification Type	INT
PatronID	Unique ID for Patron Record	INT
OrganizationID	Library	INT
NameFirst	First Name of Patron	CHAR(32)
DeliveryOptionID	Delivery Type (3,4,5 – Phone 1, 2 or 3)	INT
ReportingOrgID	Library reporting notice	INT
Amount*	Amount due on account	MONEY
CreationDate	Date notice was posted in table	DATETIME

Telephony DB

- Queue Tables – List of calls with details of items
 - ◆ TelephoneNoticeQueueTemp
 - ◆ TelephonyCalls
 - ◆ TelephonyCallDetails
- Transaction Tables
 - ◆ TelephonyCallEvents
 - ◆ TelephonyCallLogs
- Settings Tables
 - ◆ AdminAttributes
 - ◆ OrganizationSettings
 - ◆ TelephonyPatronCodeExcludes



Building the Queue

- Polaris Data dumped into temporary table
 - ◆ Build Unique PatronID list for Queue
 - If PatronID does not exist in TelephonyCalls, add it, plus add notice details to TelephonyCallDetails
 - If it does, add notice details to TelephonyCallDetails only
 - ◆ If “Call Once Per Day” is active
 - If Patron was already called, queue status is changed “Already Called”
 - ◆ Sits in Queue until next day unless cancelled.
- Queue Cleanup
 - ◆ If the notification in Polaris DB no longer exists
 - Change status of item detail to Cancelled
 - If other items exists call status remains as “Queued”
 - If all items are cancelled, change call and queue status to “Cancelled”
 - ◆ If phone number length is not 10
 - Change call status to “Invalid Phone Number”
 - Change queue status to “In-Complete”



Making the Call

- Windows Service app
 - ◆ Retrieves all calls with queue status of “Queued”
 - ◆ Sends list to Twilio
- Twilio
 - ◆ Calls patron
 - ◆ As the call progresses, Twilio sends call statuses using W-API
 - Initiated, Ringing, In-progress, Busy, Failed, No Answer
 - ◆ When call is answered, Twilio uses the W-API to retrieve message to say in XML format:
 - “Hello, this is ‘Alice’, your digital librarian from {library}, reminding you that...”



Call Cleanup

- When a call is completed or has failed
 - ◆ Windows Service app uses the Polaris API to update the notification status
 - Complete-Answered, Complete-Machine, In-complete: No answer, In-complete: Busy, Failed: Max Retries



Polaris Data Inbound using Polaris API

Field	Description
LoginBranchID	Logon branch ID
LogonUserID	Logon staff user ID
LogonWorkstationID	Logon workstation ID
NotificationStatusID	Notification Status ID
NotificationDeliveryDate	Notification Delivery Date
DeliveryOptionID	Delivery Option ID
Delivery String	Phone number
PatronID	ID of patron being notified



Polaris API Response

ID	Notification Type
0	Success
-1	Failure General
-5	Failure Database
-6	Failure invalid parameter



Twilio Cost

- After the call is complete, Windows Service Apps retrieves the final cost of the call from Twilio API and updates the call log for bookkeeping purposes
- Current rates:
 - ◆ \$0.0975 per call
 - ◆ Average calls are under a minute
 - ◆ 61 seconds is considered 2 minutes, etc.
- Inquire about Twilio.org
 - ◆ \$500 start-up credit



Available Settings

- Enable Telephony
- Enable Branch
- Number of attempts to call
- Call patron once a day
- Minutes to wait before calling
- Minutes to wait before retrying call
- Call Mail Holds
- Call Voice 1, 2, and/or 3
- Human call for holds
- Digital Voice Name
- Start/End Time for each day
- Enable Good-bye message
- Patron Codes to Exclude



Human Calls for Holds

Patron Registration - 214500 - Patron Custom Data - Polaris

File Edit View Links Tools Help

Barcode:

Registered at: Westmoreland Library Network Office (WLN)

Last name: FEE

Patron code: Adult

First name: JOSEPH

Middle name:

Expiration: Term: 0 exact date Date: 4/26/2019

Title: Suffix:

Have staff call me for holds (Phone 1, 2 or 3 preference only):

Registration

For Help, press F1

NUM



Messages

Notification Type	Message
Introduction	This is {2}, your electronic librarian from {1} with a message for {0},
Holds	we are holding {0} item/s for pickup until {1}.
Holds from another branch	{0} is holding {1} item/s for pickup until {2}.
Reminders	reminding you that you have {0} item/s that is/are soon due.
1 st Overdue	You have {0} item/s that are/is 7 days overdue.
2 nd Overdue	You have {0} item/s that is/are 14 days overdue. If this/these item/s are not returned within 15 days, you will receive a bill for replacement costs.
Trailing message for overdues	Please return these items as soon as possible.
Trailing message	For inquiry, sign into your online account by visiting my account dot w l n online dot org or call,, {0}.
Trailing message with renewals	For inquiry or to renew your items online, visit my account dot w l n online dot org or call,, {0}.

Proper Pluralization of Message

```
FUNCTION [dbo].[VerbalPluralCleanup]
( @vMsg varchar(255), @iCount int )
RETURNS varchar(255)
AS
BEGIN
    IF @iCount > 1
        BEGIN
            SET @vMsg = REPLACE(@vMsg, '/s', 's')
            SET @vMsg = REPLACE(@vMsg, 'this/', '')
            SET @vMsg = REPLACE(@vMsg, '/this', '')
            SET @vMsg = REPLACE(@vMsg, 'is/', '')
            SET @vMsg = REPLACE(@vMsg, '/is', '')
        END
    ELSE
        SET @vMsg = REPLACE(@vMsg, '/s', '')
        SET @vMsg = REPLACE(@vMsg, 'these/', '')
        SET @vMsg = REPLACE(@vMsg, '/these', '')
        SET @vMsg = REPLACE(@vMsg, '/are', '')
        SET @vMsg = REPLACE(@vMsg, 'are/', '')
    RETURN @vMsg
```



Logs and Reports

- Circulation Supervisors receive daily reports of failed or in-complete calls
- Daily log files are created on the server
 - ◆ detailing calls placed
 - ◆ when API calls are made to Polaris API with return codes
 - ◆ Cost per call



Upcoming Updates

- Group by phone number instead of Patron
 - ◆ Messages will be grouped together for multiple patrons using the same phone number to eliminate multiple calls to the same number
 - Ex: Parents with two kids (using the same phone number) will receive 1 call instead of 3 if each cardholder has a notice.
- Web UI
 - ◆ Allow circulation staff to pause Telephony while inter-delivery items are checked in at their location
 - ◆ View Queue
 - ◆ Allow circulation staff to fix phone numbers for failed calls or invalid numbers and re-queue the call
 - Will also update Patron Record via Polaris API
 - ◆ Make modifications to settings
 - ◆ View reports
- Package software for your use



Future Update

- Incoming calls for account status and phone renewals



Polaris API Resources

- Polaris API Guides
 - ◆ support.iii.com
- Polaris Developer Network
 - ◆ developer.polarislibrary.com



Polaris API Hash Sample in VB

```
Public Function Encoder(ByVal myAccessKey As String, ByVal myHTTPMethod As String, _  
    ByVal myURI As String, ByVal myHTTPDate As String, ByVal myPatronPassword As String)  
  
    Dim secretBytes() As Byte = UTF8Encoding.UTF8.GetBytes(myAccessKey)  
    Dim hmac As HMACSHA1 = New HMACSHA1(secretBytes)  
  
    Dim dataBytes() As Byte  
  
    If myPatronPassword.Length > 0 Then  
        dataBytes = UTF8Encoding.UTF8.GetBytes(myHTTPMethod + myURI + myHTTPDate + myPatronPassword)  
    Else  
        dataBytes = UTF8Encoding.UTF8.GetBytes(myHTTPMethod + myURI + myHTTPDate)  
    End If  
  
    Dim computedHash() As Byte = hmac.ComputeHash(dataBytes)  
    Dim computedHashString As String = Convert.ToBase64String(computedHash)  
    Return computedHashString  
  
End Function
```



Questions

Joe Fee

Technology Coordinator

(724) 420-5638

Joe.Fee@wlnonline.org



WCFLS Wireless Access Manager

- The objective of this project is to:
 - ◆ Replace outdated wireless access point equipment
 - Newer devices were blocking login page due to deprecated encryption protocols
 - ◆ Provide more details of deny reasons
 - ◆ Allow patrons to pay fine through Wi-Fi without seeing a desk clerk
 - ◆ Provide additional information for statistical reports



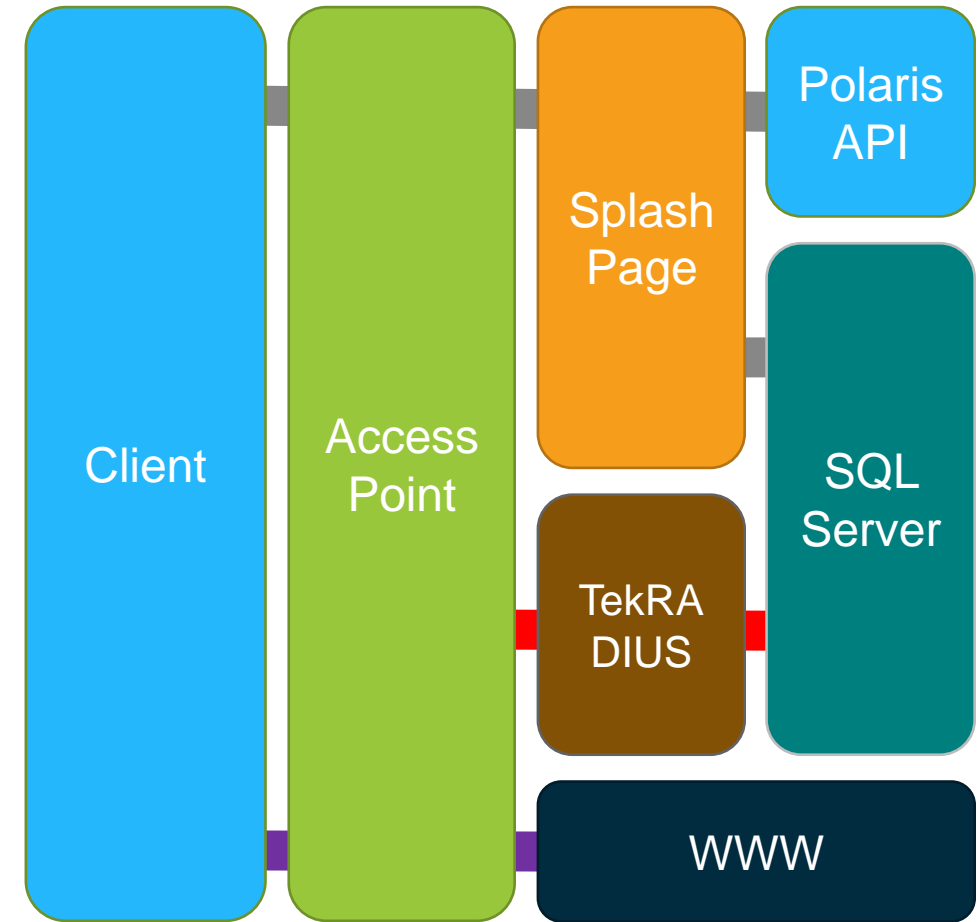
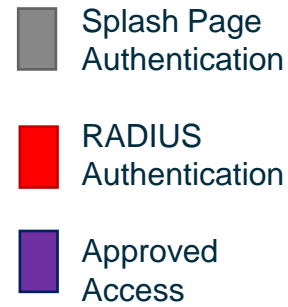
WCFLS WAM Development

- Cisco Meraki Access Points (MR32 and MR33)
- Windows IIS Server
 - ◆ Web App – developed using ASP.net in Visual Basic
- Polaris API
- WCFLS SQL Server
- TekRADIUS Server (Free Edition)

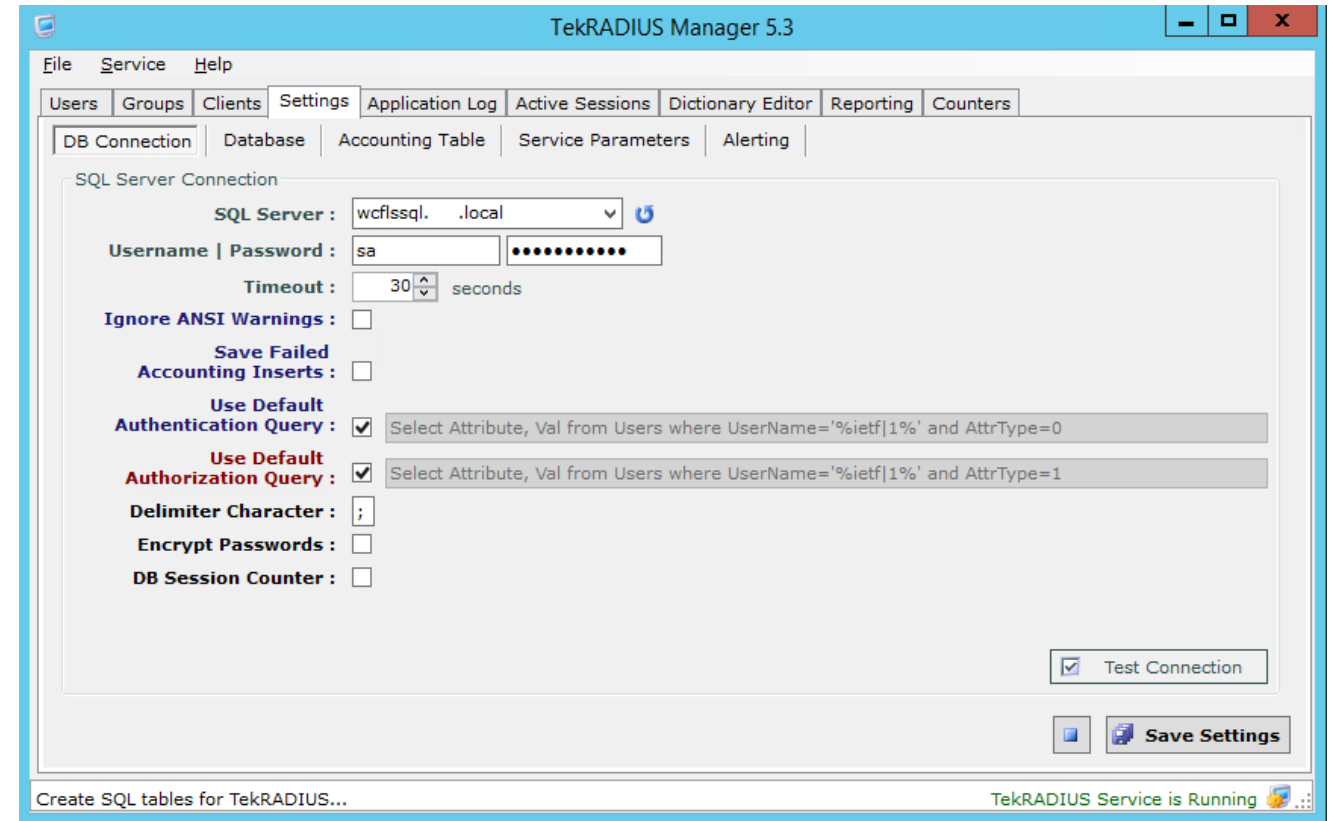


WCFLS WAM Process

1. User connects to Wi-Fi, presented with Splash Page
2. User provides library account credentials
 1. If stops exist on account, display to user
 2. If not, show Internet Agreement
3. Upon clicking “Agree”, add RADIUS User to DB
4. Wi-Fi Access Points authenticates user with RADIUS User account
5. Redirects user to original website



- SQL Server Connection
- New or Existing DB
- Tables
 - ◆ Create tables
 - Renamed with “WiFi” prepended
- Specify Secret Key



NETWORK

WLN

Network-wide

Security appliance

Switch

Wireless

Organization

Splash page

SSID:

Splash pages on this SSID are enabled because custom RADIUS authentication is enabled. You can change this setting on the [access control subtab](#).

Official themes ?

- Modern NEW
- Fluid

Custom themes ?

- Copy of Modern ✎ ✕

[Create something new](#)

Custom splash URL

- Or provide a URL where users will be redirected:

[What is this?](#)

Color customization

Color Motif:

Background Text

Content 1

Preview

- NETWORK
- WLN
- Network-wide
- Security appliance
- Switch
- Wireless
- Organization

Billing (paid access) ⓘ

Users choose from various pay-for-access options, or an optional free tier

RADIUS for splash page

#	Host	Port	Secret	Status	Actions
1	<input type="text"/>	1812	OK	↕ ✕ Test

[Add a server](#)

RADIUS accounting RADIUS accounting is enabled ▾

RADIUS accounting servers

#	Host	Port	Secret	Status	Actions
1	<input type="text"/>	1813		↕ ✕

[Add a server](#)

Enable data-carrier detect? ⓘ DCD is disabled ▾

IP addresses The Meraki cloud must be able to communicate with your RADIUS servers via the Internet.

Please make sure that:

1. Your RADIUS servers have public IP addresses (i.e., they are reachable on the Internet).
2. Your firewall, if any, allows incoming traffic to your RADIUS servers.
3. You whitelist IP addresses as clients on your RADIUS server as per the [firewall information page](#)



Westmoreland
Library Network

Welcome to the library!

Library Card Number or Username

Password

Sign into Wi-Fi →

Whoops!

Joseph, there's an issue with your library card! See below:

- \$13.00 is owed on your account. You can cancel this sign on process and use your web browser now to pay your fines through the Online Catalog by visiting myaccount.wlnonline.org or, see a Library Associate at the Circulation Desk.
- Your library card has expired, please see the Circulation Desk to review your account.

Try again ←

WLN Wifi x

Joe — □ ×

← → ↻ 🏠 🔒 Secure | https://wifi.wlnonline.org ☆ 📷 👤 ⋮

Internet Wi-Fi Policy ⋮

Acceptable Use Policy

While you are using the library's Wi-Fi network, you agree to be bound by our policies for computer use and Internet access. Any violation of this policy may result in the suspension of Wi-Fi privileges. The internet connection will be filtered in compliance with the Children's Internet Protection Act (CIPA).

Network Security

The library's wireless network is an open, public access network. Patrons should be aware that information sent wirelessly may be intercepted. Avoid entering sensitive information while using the library's Wi-Fi service, unless the web site you are visiting provides a secure connection. (A lock icon displayed in the web browser usually indicates the site is secured.)

[Agree](#)



Thank you!

