

# **Yardi Voyager**

## **Yardi Spreadsheet Reporting**

### **User's Guide**



[Corporate Website](#)

[Client Central](#)

**Documentation and Online Help**

Published by

Yardi Systems, Inc.

430 South Fairview Avenue

Santa Barbara, CA 93117

© 2013–2025 by Yardi Systems, Inc. All rights reserved.

This document is protected by copyright, trademark, and other intellectual property laws. Use of this document is subject to the terms and conditions of an authorized Yardi Systems, Inc. software license or other agreement including, but not limited to, restrictions on its use, copying, disclosure, distribution, and decompilation. No part of this document may be disclosed or reproduced in any form by any means without the prior written authorization of Yardi Systems, Inc. This document contains proprietary information about software and service processes, algorithms, and data models which are confidential and constitute trade secrets. This document is intended solely for the specific purpose for which it was made available and not for any other purpose.

The inclusion of any references to third-party products and services in this documentation does not imply endorsement by Yardi Systems of such products or services or the third party, or an affiliation between Yardi Systems and such third party. Yardi Systems is not responsible for, and does not control, any third-party products and services including, without limitation, the compatibility, performance, content, trustworthiness, legality, or any other aspect of such products and services.

**RESTRICTED RIGHTS:** This documentation is a "Commercial Item" consisting of "Computer Software Documentation," as those terms are defined in Federal Acquisition Regulation 48 C.F.R. 2.101. Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4, U.S. government end users acquire this documentation with only the rights specified in the corresponding license.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

Yardi, the Yardi logo, and all Yardi product names are trademarks of Yardi Systems, Inc.

All third-party trademarks are the property of their respective owners.

YARDI SYSTEMS, INC. HAS ATTEMPTED TO ENSURE THE ACCURACY AND QUALITY OF THE CONTENT DESCRIBED IN THIS PRODUCT OR DOCUMENT; HOWEVER, IT IS PROVIDED ON AN "AS IS" BASIS WITHOUT WARRANTY OF ANY KIND. TO THE EXTENT PERMITTED BY APPLICABLE LAW, YARDI DISCLAIMS ALL IMPLIED WARRANTIES WITH RESPECT TO THIS DOCUMENT AND THE PROCEDURES AND TECHNIQUES IT DESCRIBES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR INTENDED PURPOSE, TITLE, NON-INFRINGEMENT, COURSE OF DEALING OR COURSE OF PERFORMANCE. YARDI MAKES NO WARRANTY THAT (i) THE INFORMATION AND MATERIAL HEREIN WILL MEET YOUR REQUIREMENTS, OR (ii) THE RESULTS THAT MAY BE OBTAINED FROM THE USE OF THE INFORMATION AND MATERIALS HEREIN WILL BE ACCURATE OR RELIABLE. YARDI, ITS SUPPLIERS OR ANY THIRD PARTY MENTIONED IN THIS PRODUCT OR DOCUMENT DISCLAIM AND, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, SHALL NOT BE LIABLE FOR, ANY TYPE OF DAMAGES RELATED TO THIS DOCUMENTATION OR ITS USE, OR PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE, HARDWARE, SERVICE, OR ANY THIRD PARTY PRODUCTS AND SERVICES INCLUDING WITHOUT LIMITATION ANY OF THE FOLLOWING DAMAGES: ANY DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL, SPECIAL OR EXEMPLARY DAMAGES RESULTING FROM THE USE OF THIS DOCUMENT.

v\_ysr\_usr\_7.0.xx\_rev\_aa

## Document Changes

The following table lists the plug-in versions documented in each manual revision. To determine which plug-in versions you use, select **Administration > About**. If a manual documents multiple plug-ins, use the manual revision associated with the most recent plug-in version you use.

A number following the revision letter indicates changes since the previous revision are non-substantive: style, pagination, and so on. Thus, revision a.1 contains the same substantive material as revision a.

Publication Date	Document Revision	Newly Documented Software	Other Substantive Changes
May 2025	aa		Updated information about PublishToSharePoint parameter; added information about the ISYSRPAGEBREAKSCREENOUTPUT parameter.
January 2023	z		
March 2022	y	Correspondence Plug-in 7.13	
September 2021	x.1		
January 2021	x	Yardi Excel Add-In Five (5.004)	Updated appendices.
January 2021	w.1		
November 2020	w	Correspondence Plug-in 7.12	
August 2019	v	Yardi Excel Add-In Five	Updated appendices.
February 2019	u	Correspondence Plug-in 7.10	Added information about the YSR Admin menu; new topics on managing inactive YSR reports, downloading YSR report templates, and reviewing YSR report generation history; expanded section on Report Scheduler and the Yardi Excel Add-In.
September 2018	t.1		Added information for a new AppTask.
May 2018	t	Correspondence Plug-in 7.9	New topic on consolidated versus unconsolidated reports (report design considerations); new list of resources in Quick Start section; expanded section on Report Scheduler; new troubleshooting section.
August 2017	s	Correspondence Plug-in 7.6	

Publication Date	Document Revision	Newly Documented Software	Other Substantive Changes
April 2017	r		Added information about designing charts and graphs in Excel. Updated appendices.
January 2017	q	Correspondence Plug-in 7.3	Expanded chapter on Custom Financial Analytics reports in YSR.
June 2016	p	Yardi Excel Add-In 4.021	Reorganization and expansion of material on report section and report template setup.
May 2016	o	Correspondence Plug-in 7	
March 2016	n		Expanded information about constant values in Appendix D.
February 2016	m	Yardi Excel Add-In 4.02	Added appendices listing YSR reports and components by vertical.
December 2015	l	Yardi Excel Add-In 4.017	Reorganization and expansion of material on report setup and filtration. New material includes in-depth sections on named and numbered tokens, writing WHERE clauses, and image smart markers.
September 2015	k	Correspondence Plug-in 5.1 and Yardi Excel Add-In 4.016	
December 2014	j	Yardi Excel Add-In 4.013	New section on "Tips and Tricks" in Chapter 7, "Working With Smart Markers and Excel."
December 2014	i	Yardi Excel Add-In 4.010	Revised section on filter mapping.
November 2014	h	Correspondence Plug-in 4 and updates to the Yardi Excel Add-In	
May 2014	g	Yardi Excel Add-In	
April 2014	f	Correspondence Plug-in 3.1	
March 2014	e		Added chapter on working with smart markers in Excel.
January 2014	d		Added information about preparing templates for use with Voyager analytics. Added an index.
December 2013	c		Added information about types of custom filters.

Publication Date	Document Revision	Newly Documented Software	Other Substantive Changes
October 2013	b		Added information about Task Runner, YSR parameters, and Microsoft Word Templates.
October 2013	a		

# Contents

<b>Introduction .....</b>	<b>X</b>
About Yardi Voyager Yardi Spreadsheet Reporting .....	X
About the Documentation .....	X
Notes.....	xi
Help.....	xi
<b>Chapter 1: YSR Overview .....</b>	<b>1</b>
Introduction to YSR .....	1
Quick Start to Setting up YSR Reports .....	2
Resources .....	3
The Wizard .....	4
Report Production and Delivery .....	6
Publishing to PDF, Excel, and Word.....	7
Publishing to Screen.....	8
Merging Reports .....	8
YSR Definitions .....	9
<b>Chapter 2: YSR Report Setup .....</b>	<b>15</b>
Report Setup .....	16
Adding YSR Reports.....	17
Copying and Moving YSR Reports .....	21
Managing Inactive YSR Reports.....	21
Finding and Displaying Missing and Inactive YSR Reports.....	22
Report Design Considerations.....	23
Top-Level Select Statements and Key Columns .....	24
Identifying Your Key Column .....	25
Formatting Columns on the Report Generation Screen .....	27
Working with Consolidated Versus Unconsolidated Data.....	30
Showing Both Consolidated and Unconsolidated Data .....	35
Properties, Property Lists, and Headers for Consolidated Reports.....	37
Read-Only Databases .....	40
Voyager Analytics Report Sections.....	41
Adding Voyager Analytics Report Sections .....	42
Adding Extra Columns of Data to Voyager Analytics Report Sections .....	45
Scripted Report Sections.....	48
Adding Scripted Report Sections .....	48

YSR Scripting Conventions .....	51
Select Statements, Named and Unnamed.....	51
Session and User-Related Tokens .....	52
DDLs, DMLs, and Temporary Tables .....	54
Before and After Hooks.....	55
Example Scripts.....	56
YSR Deliverables.....	57
Troubleshooting .....	58
Error Messages .....	59
Other Problems.....	59
 <b>Chapter 3: Filtering Data with YSR .....</b>	 60
Filtration Overview .....	60
Classic Versus New Filter View .....	61
YSR Report Filter Definition .....	62
YSI Property Lookup Lists.....	62
Defining YSR Report Filters .....	65
Defining Filter Fields for Use with Voyager Analytics .....	69
Defining Filter Fields for Data Range.....	70
Defining Dynamic Lists and Custom Lookups .....	72
Displaying User's Filter Criteria in Title of Report (Dynamic Titles) .....	75
Filter Mapping for Voyager Analytics Report Sections.....	77
Filter Mapping for Scripted Report Sections .....	81
Named Tokens Versus Numbered Tokens .....	82
Named Token Example.....	84
Numbered Token Example .....	84
Filtering the Top-Level Select Statement.....	86
Writing WHERE Clauses.....	87
 <b>Chapter 4: YSR Report Section and Template Setup .....</b>	 89
Report Section and Template Setup Overview .....	89
Report Section Example .....	90
Excel Template Example.....	91
Word Template Example .....	93
Nested Tables in Word .....	94
Working with Word 2013 .....	99
PDF Template Example.....	100
Report Output Example.....	101
Report Section Setup .....	101
Setting up Report Sections .....	101
Adding Static Attachments to Report Sections.....	105

Report Template Setup .....	108
Identifying Field Names for Voyager Analytics Reports .....	108
Adding Images to Word Templates.....	111
Displaying Text Containing HTML Tags in Word Templates.....	112
Downloading YSR Report Templates .....	113
Dynamic Template Selection .....	114
Selecting Templates According to Criteria in Top-Level Select .....	115
Selecting Templates According to User's Filter Criteria .....	116
Selecting Templates Using Other Runtime Options .....	118
<b>Chapter 5: Production and Delivery .....</b>	<b>120</b>
Email, Attachments, and Publication.....	120
Attaching Reports .....	122
Attaching and Emailing Reports.....	124
Publishing to SharePoint.....	129
Publishing to Portal .....	131
Holding Emails .....	131
Static PDF Attachment Error Logs.....	132
Monitoring Reports.....	132
Reviewing YSR Report Generation History .....	133
Output Options .....	134
Displaying Graphical Elements on Screen .....	134
Generating Ad Hoc Reports in Report Scheduler.....	137
Generating QR Codes Dynamically at Runtime.....	139
Digital Signatures.....	141
Setting up Digital Signatures .....	144
Adding Digital Signatures to YSR Reports .....	146
Adding Digital Signature Tokens to Report Templates.....	147
Task Runner .....	148
Report Scheduler .....	151
Preparing a Report Scheduler Script .....	152
Example Report Scheduler Script.....	153
Scheduling a YSR Report that Lacks a Top-Level Select.....	154
Scheduling Your Report.....	155
Monitoring Reports Generated with Report Scheduler .....	157
Side Menus .....	158
Adding Custom Menu Paths to YSR Reports in Voyager.....	158
Automatically Generating YSR Reports from a Side Menu.....	160
Adding Menu Paths for YSR Reports in VoyagerPlus .....	161
Adding the YSR Admin Menu .....	162

Custom Help.....	163
<b>Chapter 6: Custom Financial Analytics Reports in YSR.....</b>	<b>164</b>
CFA Report Setup Overview.....	165
Setting Up CFA Reports in YSR.....	165
Generating Consolidated Data .....	167
Adding Top-Level Selects for Consolidated Reports.....	168
Formatting CFA Reports .....	169
CFA Account Tree Reports in YSR .....	170
Account Tree Report with Bold and Underline.....	170
Account Tree Report with Manipulated Data .....	176
CFA Property Portfolio Reports in YSR.....	181
Property Portfolio Report, Summarized by Attribute .....	182
Property Portfolio Report, Summarized by Property .....	186
CFA Comparison Reports in YSR .....	187
Property Comparison Reports in YSR.....	187
Attribute Comparison Reports in YSR .....	194
<b>Chapter 7: Working With Smart Markers and Excel .....</b>	<b>202</b>
Smart Marker Overview.....	202
Smart Markers and Excel Data Types .....	203
Smart Marker Parameters .....	204
Supported Parameters.....	204
Using Parameters to Display Data in Alternating Rows .....	205
Excel Formulas and References.....	206
Example Formula - Running Balance .....	207
Formula Syntax and Parameters .....	207
Troubleshooting Broken Excel References .....	209
Troubleshooting Formulas That Display No Data.....	210
Groups and Subtotals .....	210
Data Groups.....	211
Subtotaling a Single Data Group.....	212
Subtotaling Multiple Data Groups.....	213
Labeling Subtotals .....	215
Grand Total Tricks.....	218
Images .....	220
Details (Show, Hide, and Delete with YSROptions) .....	222
Charts, Graphs, and Pivot Tables.....	227
Adding Charts, Graphs, and Pivot Tables .....	227
Tips and Tricks .....	233
Adding Special Formatting to Expanding Row Data .....	233
Date and Number Formatting .....	235

Freeze Panes.....	237
Drill-Down Links .....	238
Displaying Data Horizontally.....	239
<b>Chapter 8: Yardi Excel Add-In for YSR .....</b>	<b>242</b>
Yardi Excel Add-In Overview .....	243
Building YSR Report Templates with the Add-In .....	244
Downloading and Installing the Yardi Excel Add-In .....	245
Downloading YSR Design Helper.....	248
Setting the Default YSR Design Helper Path .....	249
Enabling the YSR Smart Marker Designer.....	250
Selecting Data Sources .....	251
Creating Smart Markers with the YSR Smart Marker Designer.....	254
Creating YSR Hyperlinks with the YSR Hyperlink Designer .....	258
Checking Your Excel Template for Errors .....	262
Checking Your Excel Template for Conditional Format Rules .....	263
Building Merge Token Files for Word Templates .....	264
Viewing YSR Analytic Filter Usage Documentation.....	265
Building YSR Reports with the Add-In (Dump SQL) .....	267
Preparing Script Files For Use With the Yardi Excel Add-In .....	268
Adding a Top-Level Select Statement .....	269
Building the //Filter Section.....	270
Filter Mapping and the Value1 Parameter.....	272
Val1 Parameter Conventions.....	272
Building Report Packages (Dump SQL) .....	273
<b>Appendix A: YSR Report Schema .....</b>	<b>279</b>
<b>Appendix B: YSR Reports and Component Files by Vertical .....</b>	<b>280</b>
<b>Appendix C: YSR Reports with Voyager Analytics Data Sources .....</b>	<b>402</b>
<b>Appendix D: Constant Values for Analytics Data Source Filter Mapping .....</b>	<b>408</b>

# Introduction

## About Yardi Voyager Yardi Spreadsheet Reporting

With Yardi Voyager Yardi Spreadsheet Reporting (YSR), you can generate reports using formatting and graphing features from the Microsoft Office suite.

This manual includes information current through Voyager 7S Correspondence Plug-in 7.13.



When upgrading to Voyager 7S Correspondence Plug-in 7.12 or later, contact Yardi technical support to ensure that the Global Assembly Cache (GAC) installer (the version present in the Misc sub folder of Correspondence Plug-in 7.12) has been rerun on all your organization's Service Manager nodes.

## About the Documentation

This document describes how to set up and use YSR.

Your system administrator customizes YSR and manages the security settings. Customization and security settings affect the appearance of Yardi Spreadsheet Reporting and determine the options that are available to users. Most screens and menu paths described in this document are for a standard implementation with the least restrictive security settings. The screens and descriptions may not match those that you see when you use YSR. This document does not typically describe fields like **Name** or **Unit #**, for which the purpose is self-evident. Unless otherwise indicated, all menu paths start from the Voyager System Administration side menu.

You can find the latest documentation on Yardi [Client Central](#):

<https://clientcentral.yardi.com>

If you need help determining your logon name and password, contact Yardi technical support.

You can use the following documentation with YSR:

Document	Describes
<i>Core Administration Guide</i>	Installation and administration tasks common to all Voyager systems.
<i>Core Setup Guide</i>	Setup tasks common to all Voyager systems.
<i>Core User's Guide</i>	Use of features common to all Voyager systems.
<i>SQL Scripting User's Guide</i> (Archived)	How to author SQL scripts, some aspects of which are applicable to YSR reports that use custom SQL data sources.

Document	Describes
User's guide for each Voyager system, module, or utility	Installation, setup, and use of a Voyager system, module or utility.
Procedures guides for Voyager Residential, Voyager Commercial, and Accounting	How most users typically perform common functions. Available in Microsoft Word format for user customization.

## Notes



An information note provides background information. For example, it may explain how changes made in one screen affect data that appears in another screen.



A caution note explains how to avoid a potential problem, or indicates that a process will cause irreversible changes to your data.



A tip describes a way to get more from your software. For example, it may explain an alternative way to perform a task.

## Help

Most Voyager screens have a **Help** button for quick access to information about using the screen.

# CHAPTER 1

## YSR Overview

### In this chapter:

Introduction to YSR .....	1
Quick Start to Setting up YSR Reports.....	2
Report Production and Delivery .....	6
YSR Definitions .....	9

This chapter provides a basic overview of YSR and defines the elements that make up a YSR report.

## Introduction to YSR

YSR is the most flexible and customizable Voyager reporting technology that Yardi offers. With YSR, you can create presentation-quality, client-deliverable reports and report packets using templates designed in MS Excel and MS Word. You can take advantage of all the formatting and graphing capabilities of Excel and Word when designing YSR reports, including pivot tables and graphs in Excel. You can also retrieve data from anywhere in your database with YSR, since YSR lets you add custom SQL sections to your report design. You can also retrieve data for YSR reports by incorporating Voyager analytics data, making YSR an ideal tool for customizing Voyager analytics reports.



You can create both single reports and report packets with YSR. This document uses the term *YSR report* to refer to all types of reports created in YSR, whether the report contains one or many sub-reports.

Every YSR report has at least one sub-report. Sub-reports can use either or both of YSR's data retrieval methods:

- Voyager analytics
- Custom SQL scripts

If you are writing custom SQL scripts to retrieve data, you can customize your script to retrieve any data in your database. You can also use YSR to customize report data associated with many standard Voyager analytics. You can reformat Voyager analytics reports, and in some cases you can add associated column data to analytics reports for use by YSR.

Whether you are using custom SQL, Voyager analytics, or both, YSR gives you tools to modify both the content and format of your reports.

You can customize the formatting of all YSR report data by modifying your *report templates*. Report templates are Word or Excel documents that contain the formatting and layout specifications for a report. Every sub-report in a YSR report has its own report template. You can use all the native power of Word and Excel when designing your report templates, including all the formulas, graphs, charts, and pivot tools available in Excel.

You can publish YSR reports to screen, Excel, PDF, and Word. You can send YSR reports to your contacts as attachments, and you can publish YSR reports to SharePoint and Investor Portal. YSR integrates fully with Task Runner, and you can use YSR across all Voyager modules.

### Reports powered by Voyager analytics

Voyager analytics are the standardized reports that generate data on screen in Voyager. You can find them organized under menu links such as Financial Analytics, AR Analytics, AP Analytics, and Commercial Analytics. While you can generate these reports without using YSR, there are benefits to using them with YSR. With YSR, you can:

- Add data to, and subtract data from, analytics report data.
- Modify the formatting of your reports by editing your templates in Word or Excel.
- Create custom filters to alter the options available to users when generating reports.

The clear advantage of using analytics reports in YSR is that you do not need to know SQL to set up custom reports. You can also be sure that your data ties out against reports that you generate with Voyager analytics because you are using the same report engine.



Not all Voyager analytics are available for use with YSR, but the number of analytics integrated into YSR is increasing.



For more information about Voyager analytics reports, as compared to standard reports, see “Reports” in the *Voyager Core User’s Guide*.

## Quick Start to Setting up YSR Reports

### In this section:

Resources .....	3
The Wizard .....	4

Yardi provides many resources to support YSR report writers. New report writers will benefit from all of the resources described here.

For the quickest overview of all YSR report design essentials, see “*Learning YSR in 10*” on Client Central. This 2-part video covers all the fundamentals of report setup, including how to work with Analytics-based report sections and scripted report sections, how to design a top-level select and key column, and how to email and attach reports.

## Resources

This section lists resources available for new YSR report writers.

### Client Central Videos

See “*Learning YSR in 10*” on Client Central. This 2-part video covers all the fundamentals of report setup.

### YSR Newsletters

Find YSR Newsletters on Client Central, accompanied by the YSR Newsletter Topic Index. The Newsletters cover technical topics, tips and tricks, best practices, and solutions to complex report design problems. Newsletters date from November 2014 forward.

### YSR Menu Sets

YSR Menu Sets is the name of a zip file available on Client Central. The file contains a PKG file that, on execution into your database (through Voyager Workstation Administrator), adds two menu sets to your database: YSRAll and YSROur. These menu sets provide organized menu links to the setup and generate pages for all example reports (YSR) and the example reports appropriate to your plug-ins (YSROurs), including any custom reports in your database. Use these menu sets to more easily access and peruse the reports already in your database.

### Yardi eLearning

Yardi eLearning offers a series of YSR classes that cover all the basics of YSR report design. Topics include:

- All the Essentials in 10 Minutes
- Scripted Report Setup, Step-by-Step
- Analytics-Based Report Setup, Step-by-Step
- YSR and Excel: Graphs, Totals, Grand Totals
- Custom Financial Analytics in YSR (Account Tree Reports)
- Custom Financial Analytics in YSR (Portfolio Reports)
- Custom Financial Analytics in YSR (Comparison Reports)
- One vs. Many (Header/Details, Consolidation Options)

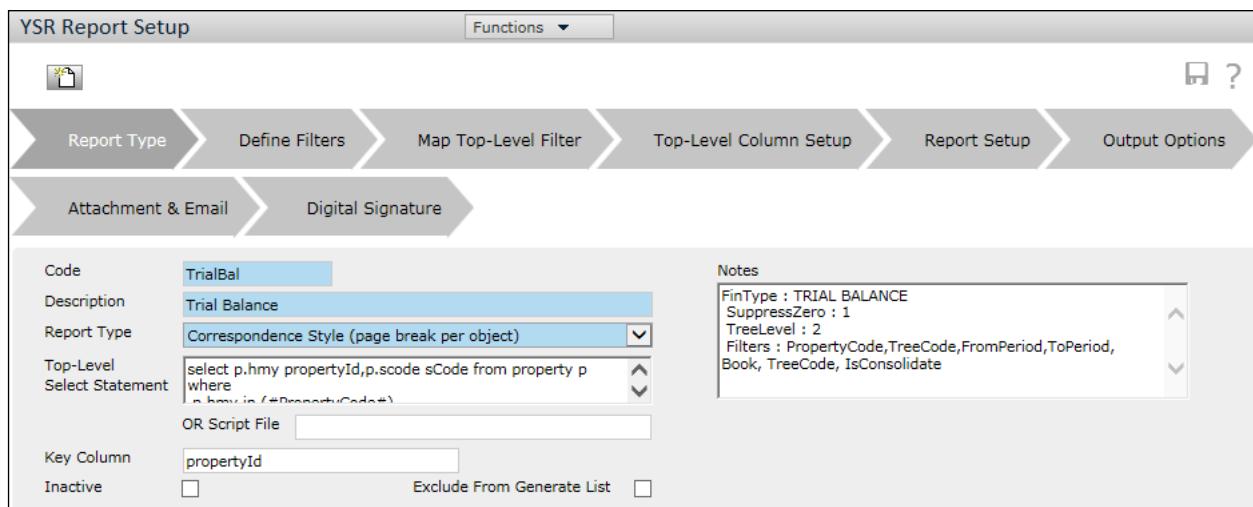
## YASC Courses

YASC offers a wide range of introductory and advanced topics for YSR report writers. Attend in person or find the supporting PowerPoint slide decks on Client Central.

- RE060 Introduction to Yardi Spreadsheet Reporting
- RE540 Yardi Spreadsheet Reporting Technical Session I
- RE541 Yardi Spreadsheet Reporting Technical Session II
- RE250 Using YSR for Custom Financial Analytics Reports
- RE405 Reporting Comparison between YSR and SSRS
- RE575 Transitioning from Crystal to SSRS & YSR

## The Wizard

With Correspondence Plug-in 7.9, you can use the YSR Report Wizard to help design reports. The Wizard offers a sequential approach to report-design, with tabs for each part of the report design process. For example, the YSR Trial Balance report is shown in the Wizard in the following graphic:



The Wizard is turned on by default after installation of Correspondence Plug-in 7.9. To turn the Wizard off or back on, see "To turn the Wizard on and off" on page 5.

The Wizard contains all the same screens and screen elements as the classic YSR report setup screens, and all report design concepts are the same. The difference is the sequential layout of screens and the addition of some user experience improvements like better error messaging.

Because all screen elements are the same, the documentation does not include Wizard-specific screen definitions or procedures.

### To turn the Wizard on and off

- 1 Select **Admin > YSR Correspondence > Setup Report**. The **YSR Correspondence Setup Screen** appears.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see “Adding the YSR Admin Menu” on page 162.

YSR Correspondence Setup Screen	
YSR Report Code	(dropdown menu)
Inactive?	No
Wizard?	Yes
Display Rows	300
<b>Submit</b> <b>Clear</b> <b>Help</b>	

- 2 Edit the **Wizard** field (select **No** to turn the Wizard off).
- 3 Click **Submit**. The **YSR Report Setup** screen appears in the mode you selected. Voyager remembers your selection.

# Report Production and Delivery

## In this section:

Publishing to PDF, Excel, and Word .....	7
Publishing to Screen .....	8
Merging Reports.....	8

When you set up a YSR report, you also set up the production and delivery options available to users. The following table describes the report destinations you can choose from, depending on the type of report templates in your YSR report.

	YSR Report with Multiple Sub-Reports			YSR Report with a Single Sub-Report		
Report Destination	Excel Templates	Word Templates	PDF Templates	Excel Template	Word Template	PDF Template
Excel	X			X		
PDF	X	X	X	X	X	X
Word		X			X	
Screen	X			X		



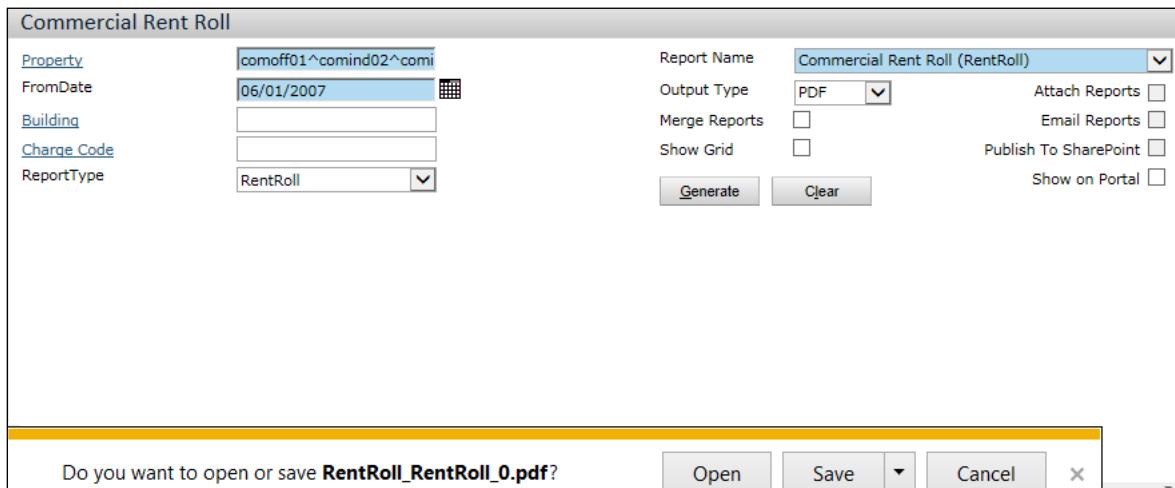
Some Excel reports with highly complex graphs or pivot tables may not have satisfactory formatting when generated on screen. In such cases, generate the report in Excel.

You can deliver all reports to contacts via email, you can attach reports to Voyager objects, and you can publish reports to SharePoint and Investor Portal. For in-depth information about email and document management options, see “Troubleshooting” on page 58.

## Publishing to PDF, Excel, and Word

If users choose to publish the report to PDF, Excel, or Word, you can set up the YSR report to display the reports in two different ways:

- You can publish reports directly to PDF, Excel, or Word. The following graphic shows the pop-up window that appears in Internet Explorer when YSR generates a report directly to PDF.



- You can use a top-level select statement to organize your reports on screen with links to the report files. The following graphic shows a YSR report that uses a SQL select statement to organize reports on screen with links to the reports.

The screenshot shows an 'IM report\_transaction history by date' report configuration window. On the left, there are filter fields: 'Investor code' (yfund), 'Investment code' (yfund), 'fromdate' (01/01/2008), and 'todate' (12/31/2013). On the right, there are report settings: 'Report Name' (IM report\_transaction history by date (IM\_rep1)), 'Output Type' (Excel), 'Merge Reports' (unchecked), 'View' (checked), 'Show Grid' (unchecked), and buttons for 'Generate', 'Clear', 'Attach Reports' (unchecked), 'Email Reports' (unchecked), 'Publish To SharePoint' (unchecked), and 'Show on Portal' (unchecked). Below the settings, a table displays transaction history data:

investorhmy	investorcode	investornname	Report
143.00	yinv1	Investor1	<a href="#">View Report</a>
144.00	yinv2	Investor2	<a href="#">View Report</a>
145.00	yinv3	Investor3	<a href="#">View Report</a>
146.00	yinv4	Investor4	<a href="#">View Report</a>
147.00	yinv5	Investor5	<a href="#">View Report</a>



For more information about organizing your report results, see "Formatting Columns on the Report Generation Screen" on page 27.

## Publishing to Screen

If your report contains Excel templates only, you can give users the option to publish it directly to screen. The following graphic shows a commercial rent roll for a single property published to screen in YSR.

Property	Unit(s)	Lease	Lease Type	Area	Lease From	Monthly Rent
Sunrise Tower,Santa Monica	350A, 350B, 350C, 350D	Roxio, Inc.	Office - Net	13,380.00	37,288.00	51,513.00
Sunrise Tower,Santa Monica	550A, 550B	Law Offices of Wallen & Klarek	Office - Net	6,084.00	38,078.00	22,693.32
Sunrise Tower,Santa Monica	250A, 250B	Beeker Group Architects	Office - Net	2,484.00	38,169.00	9,290.16
Sunrise Tower,Santa Monica	650C, 650D	London & Parish, CPA	Office - Net	3,672.00	37,926.00	12,668.40
Sunrise Tower,Santa Monica	450A, 450B	Sterk, Cooke & O'Neill, CPA	Office - Net	4,574.00	37,895.00	17,106.76
Sunrise Tower,Santa Monica	850D	Truman Agency	Office - Net	3,166.00	37,773.00	12,885.62
Sunrise Tower,Santa Monica	750C	Kirkland Systems	Office - Net	2,122.00	37,653.00	8,275.80

If the user filters for multiple properties, additional data rows appear. You can also publish graphs, charts, and tabbed data to screen. For more information about screen output options, see “Output Options” on page 134.



- When publishing to screen with a page-break parameter, use ISYSRPAGEBREAKSCREENOUTPUT.
- Use the PageBreakYSRParameter for publishing to PDF, Excel, or Word.

## Merging Reports

One advantage of YSR is its capacity to merge multiple instances of a report into a single document. For example, the following graphic shows multiple reports merged into one Excel document.

A	B	C	D	E	F	G	H	I	J	K	L
9											
10	Total Commitment						\$		-		
11	Less: Cash Capital Contributions						\$	(3,100,200)			
12	<b>Remaining Capital Available to be Called</b>						\$	(3,100,200)			
13											
14	<b>Total Capital Called</b>						\$	(3,100,200)			
15	Less: Total Distributions to Date								-		
16	<b>Total Capital Outstanding</b>						\$	(3,100,200)			
17											
18	Date						Contribution		Distribtuion		
19											
20	03/31/14						\$ (100)	\$	-		
21	04/30/14						(3,000,000)		-		
22	05/01/14						(100,100)		-		
23											
24	<b>Total</b>						\$ (3,100,200)	\$	-		
25											
26											
27											
	IM_rep1-59 / IM_rep1-60 / IM_rep1-69 / IM_rep1-143 / IM_rep1-144 / IM_rep1-145 / IM_rep1-146										

In this example, the pre-rendered Excel report template has unnamed worksheets. When the user generates the report, Voyager renames each worksheet using a combination of the report code (IM\_rep1 in this example) and the page break column (property hmvs, in this example).



In order to merge reports, you must use a top-level select statement and key column. For more information about top-level select statements, see “Report Design Considerations” on page 23.

Depending on report design, YSR can also merge only a subset of reports. For example, in the following graphic, YSR merges only those reports whose intended recipients are missing email addresses.

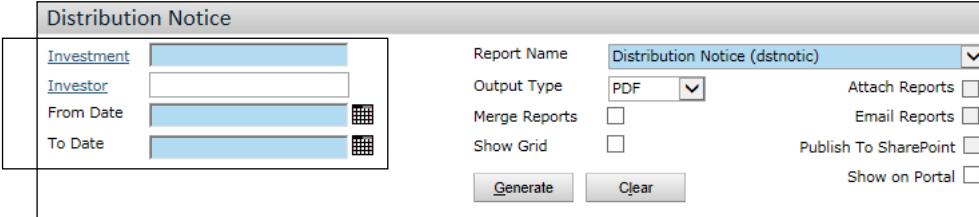
OwnerCode	Owner	PropCode	Property	PropertyID	OwnerID	Report	Email
glegal	Mike Delia	gprop1	Triangle Town Center	21,00	72,00	View Report	View Email
glegal	Mike Delia	gprop2	mutha chambers	22,00	72,00	View Report	ERROR: Recipient contact not setup.
glegal	Mike Delia	p0000006	crompton	170,00	72,00	View Report	ERROR: Recipient contact not setup.



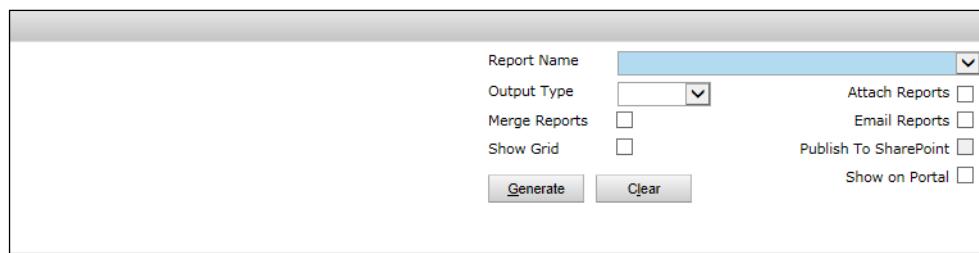
For more information about setting up YSR reports to merge documents with missing emails, see “Attachment & Email Screen Reference” on page 128.

## YSR Definitions

This section defines the terms used in the document to refer to various YSR elements.

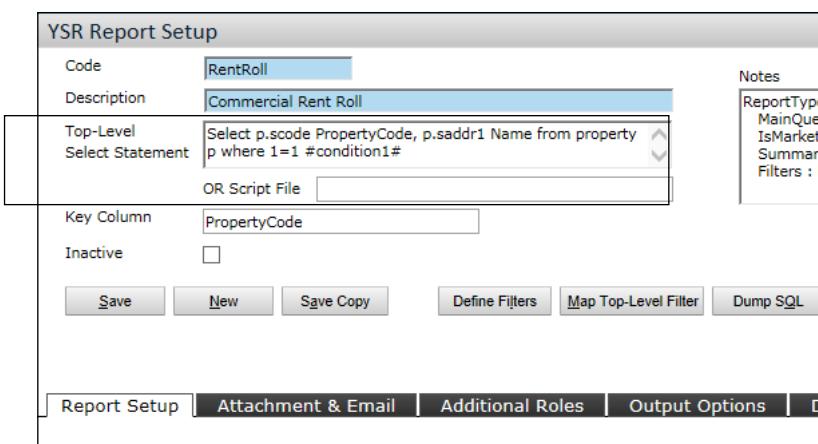
Term	Definition
analytics report	See <i>Voyager analytics</i> .
custom filter fields	Filter fields that appear on the report generation screen when a user selects a YSR report to generate (the YSR report filter). Every YSR report has its own set of custom filter fields.
	
	<p>You must define the custom filter fields for every YSR report. You can access the setup screen for custom filter fields by clicking <b>Define Filters</b> on the <b>YSR Report Setup</b> screen.</p> <p>For more information, see “Defining YSR Report Filters” on page 65.</p>
Custom Correspondence	A legacy name for YSR. A related term ( <code>CustomCorrespMerging</code> ) is the name of the root table for YSR reports in your database.
custom token	Placeholder for Voyager data. You can create custom tokens to add additional columns of data to some Voyager analytics-based reports. For more information, see “To add a Voyager analytics report section” on page 43.
data column	Structural element of a data table. Each column in a table contains a different type of data. For example, a data table containing property information might contain one column for property codes, one column for property addresses, one column for the property’s primary contact person, and so on.
Dump SQL	<p>Button that downloads a Voyager .pkg file to the user’s computer for migration of the YSR design to another SQL Server database. You can access this button on the <b>YSR Report Setup</b> screen.</p> <p>A YSR .pkg file contains the complete definition of a YSR report, including filter definitions, filter mapping, report section setup, and definition of data sources. A YSR .pkg file does not contain the report templates or any custom SQL scripts associated with the YSR report.</p> <p>YSR .pkg files are generally compatible only when migrating between SQL Server databases that are at the same Correspondence plug-in level. Since the database schema may evolve as plug-ins are installed, a package that was generated from a Voyager webshare and then upgraded to a higher plug-in level will likely fail if run against a database at a lower plug-in level. The newer database schema will include field definition inserts or updates not present in the older schema.</p>
filter conditions	Section of a SQL select statement that defines the filter criteria that data records must meet to be included in results. Yardi scripts typically contain filter conditions in the WHERE clauses of select statements. With YSR, you can hard-code your filter conditions into custom SQL scripts or you can leave placeholder tokens in your script and use YSR filter mapping screens to manage filter conditions. For more information about filtration, see Chapter 3, “Filtering Data with YSR.”

Term	Definition
foreign key	<p>Column in a relational data table that points to a primary key in another table.</p> <p>In Yardi schema, foreign keys usually start with <b>h</b> followed by the name of the related table. For example, the Tenant table has a foreign key column called hUnit. The hUnit column points to the primary key of the Unit table which, in accordance with the conventions of Yardi schema, is named hmy.</p>
key column	See <i>primary key</i> .
list	<p>A set of values that users can select from when completing the YSR report filter at run-time. You can create a basic list using carets (^) to separate list values (North^South^West^East), and can write a dynamic list using a short SQL statement. For more information about adding lists, see “Defining Dynamic Lists and Custom Lookups” on page 72.</p>
lookup list	<p>A type of filter field that offers the user a pop-up window where they can browse or search through values in their database.</p> <p>When you build the YSR report filter, you can use standard YSI lookup lists like ysiPropertyOrListLookup, or you can write your own custom lookup lists. You must write two SQL select statements to create a custom lookup list.</p> <p>For more information, see “Defining Dynamic Lists and Custom Lookups” on page 72.</p>
merge field	<p>Placeholder for Voyager data in a Word template. Merge fields must match the field aliases in the SQL select statement used to retrieve data.</p> <p>For more information about merge fields, see “Word Template Example” on page 93.</p>
Merged Report	A legacy name for YSR, which highlights the fact that a YSR report can contain many merged sub-reports.
page break column	<p>Database column containing the values Voyager uses to label merged reports when published to Excel.</p> <p>The page break column must point to a data column that uniquely identifies each reporting object (like the hmy or scode column of the property table). To use the page break column to manage merged report labels, you must use a top-level select statement and identify the key column of each report section.</p> <p>If you leave the page break column blank, Voyager retains the worksheet name used in the Excel report template.</p> <p>If you supply a page break column, Voyager uses this convention:</p> <ul style="list-style-type: none"> <li>• Template with named worksheets: WorksheetName-KeyColumnName</li> <li>• Template with unnamed worksheets: ReportCodeValue-KeyColumnName</li> </ul> <p>Applicable to reports that use a top-level select statement and key column only.</p>
primary key	<p>Column in a relational data table that uniquely identifies each record in the table.</p> <p>In Yardi schema, primary keys in data tables are usually called hmy, pronounced myhandle. Voyager populates the hmy column by auto-numbering. For example, the Unit table has a primary key of unit.hmy. When this handle appears as a foreign key in a related table, it is usually named as a contraction of h and the table name. So for instance, the Tenant table has an hUnit field. hUnit is a foreign key referring back to the primary key of the Unit table.</p> <p>The terms <i>primary key</i>, <i>key column</i>, and <i>relational key</i> appear interchangeably in the documentation.</p>

Term	Definition
relational key	See <i>primary key</i> .
report code	A unique code for a YSR report (the scope of the YSR report). The code can be up to 16 characters long. You set up the report code in the <b>Code</b> field of the <b>YSR Report Setup</b> screen. This code is the unique identifier of the YSR report, and it is used when you move a YSR report from one database to another (via Dump SQL).
report generation screen	Screen where users can generate YSR reports. The screen does not have a screen name until the user selects a YSR report from the <b>Report Name</b> drop-down list. Then Voyager provides the name of the YSR report as the screen name.
	<p>System Administration menu: <b>Admin &gt; YSR Correspondence &gt; Generate Report</b></p> <p><b>YSR Admin</b> side menu: <b>Generate Report &gt; Generate Report</b></p> <p><b>DBO Manager</b> screen: <b>YSR Correspondence &gt; Generate Report</b></p> 
report section	The most granular component of YSR report design, corresponding to a single named SELECT or a single Voyager analytics data source.
	<p>Every sub-report in a YSR report has at least one section.</p> <p>For more information, see “Report Sections Setup Screen Reference” on page 103.</p>
report template	<p>Excel or Word document containing text and formatting for a sub-report. Excel templates use smart markers and Word templates use merge fields to represent Voyager data. When users generate the YSR report, Voyager replaces the smart markers and merge fields with Voyager data to create the report output.</p> <p>Voyager retrieves templates from the Reports path specified on the <b>Accounts and Options</b> screen. For more information about the Reports path, see “Web Environment” in the <i>Voyager Core Administration and Setup Guide</i>.</p>
section code	A short code for a report section, used in Excel smart markers.

Term	Definition
select section name	The name of a select statement in a SQL script or Voyager analytics report.
	<pre>//SELECT KPIData SELECT     p.SCODE          PropCode     ,CommKPI.DTDATE  KPIDate     ,CommKPI.DTOTALAREA TotalArea     ,CommKPI.DOCCUPIEDAREA OccupiedArea     ,CommKPI.DLEASEDAREA LeasedArea FROM     ASCAP_CommKPI AS CommKPI     INNER JOIN PROPERTY AS p ON CommKPI.HPROP = p.HMY WHERE 1=1     AND CommKPI.DTDATE = dbo.perf_eop('#AsOfMonth#', 'm')     #Conditions# //END SELECT</pre>
	<p>When you use a custom SQL script to retrieve data for one or more YSR sub-reports, you must give a unique name to each select section in the script, regardless of whether there are one or many SELECT sections in the same script. You must associate each named select section with a YSR report section on the <b>Report Sections Setup</b> screen.</p>
	<p>Top-level select statements have slightly different requirements.</p>
	<p>If you store your top-level select statement in a text file and reference the file from the <b>YSR Report Setup</b> screen (rather than authoring the top-level select statement directly on screen), you must leave the top-level select section unnamed. The script file must contain only one unnamed select section. Voyager interprets the unnamed select section as the top-level select statement.</p>
	<p>The line that begins a select section must start with //SELECT, followed by a single space, followed by the name of the select. No other text, such as comments or trailing spaces should subsequently appear on this line. The name can be a maximum of 100 characters long, and should not include embedded spaces, embedded numbers, or any special characters. The only exception to these restrictions is that the name may end with numeric characters. For example, "Select42" is valid, but "Select43A" might not function as expected. The entire line is processed as case insensitive.</p>
SELECT statement	<p>Portion of a SQL script that retrieves a set of records from one or more data tables. YSR scripts must begin with //SELECT and end with //END SELECT.</p>
smart marker	<p>Placeholder for Voyager data in an Excel template. Smart markers begin with &amp;= followed by the select section code and the field alias used to retrieve the data. For example, &amp;=co.Investorname represents the field InvestorName in the dataset retrieved by an execution of the select section coded co.</p>
	<p>Smart markers expand dynamically to accommodate multiple rows of data.</p>
	<p>For more information about using smart markers, see Chapter 7, "Working With Smart Markers and Excel."</p>
SQL script	<p>A text file that obeys the structural rules of Yardi SQL Scripted text files, but requiring only //SELECT... //END SELECT sections. For more information, see "YSR Scripting Conventions" on page 51.</p>

Term	Definition
standard analytics report	<p>Appears on the <b>Report Sections Setup</b> screen; refers to Voyager analytics reports and a few additional data sources like IM data and Performance data.</p> <p><b>NOTE</b> Do not confuse standard analytics reports, as used on YSR screens, with Voyager standard reports. Voyager standard reports are scripted reports. Standard analytics reports in YSR include Voyager analytics reports that are exposed to YSR.</p>
	<p>For more information about the difference between Voyager standard reports and Voyager analytics reports, see “Report Types” in the <i>Core User’s Guide</i>.</p>
sub-report	<p>Individual report contained in a YSR report, corresponding to one Excel or Word report template. Every sub-report must have at least one report section, corresponding to one data retrieval method.</p>
template	<p>See <i>report template</i>.</p>
top-level select statement	<p>Select statement that determines how results appear on the report generation screen. Optional if your YSR report contains just one Excel-based sub-report. If your YSR report contains multiple sub-reports or any Word templates, the top-level select statement is required.</p> <p>You can enter the top-level select statement into the <b>Top-Level Select Statement</b> field on the <b>YSR Report Setup</b> screen or you can save it as a .txt file and enter the file name in the <b>OR Script File</b> field.</p>



For more information, see “Report Design Considerations” on page 23.

Val 1 parameter	<p>Value 1 parameter. The string that follows WHERE in the WHERE clause of a SQL select statement. For example, <b>p.hmy in (#property#)</b> is a possible Val 1 parameter.</p>
	<p>For more information, see “The Value 1 Parameter” on page 81.</p>
unique identifier	<p>Element in a data table that refers to just one data record. The unique identifier appears in the key column of the table. In Yardi schema, the unique identifier often appears in the hmy column.</p>
Voyager analytics	<p>Standardized reports, published by Voyager vertical or module, that generate data on screen in Voyager. You can use some Voyager analytics to retrieve data for YSR reports. <i>Voyager analytics</i> and <i>analytics reports</i> appear interchangeably in the documentation.</p>
	<p>The <b>Report Sections Setup</b> screen refers to Voyager analytics data sources as standard analytics reports.</p>

Term	Definition
Yardi SQL script file	<p>A structured text file, using the conventions described in the <i>Voyager SQL Scripting Guide</i>.</p> <p>When generating a YSR report, Voyager processes only the //SELECT... //End SELECT statements. You can therefore reuse the select sections of existing Yardi SQL script files within YSR. To reuse an existing script file, you must recreate the custom filter fields defined in the script by clicking <b>Define Filters</b> on the <b>YSR Report Setup</b> screen. For more information, see “Defining YSR Report Filters” on page 65.</p>
YSR report	<p>Any report created in YSR, whether the report contains one or many sub-reports.</p> <p>When you generate a YSR report, Voyager inserts a unique record into the CustomCorrespMerging table.</p>
YSR report filter	<p>The custom filter fields that appear to users when they generate a YSR report. Every YSR report has its own report filter, or set of custom filter fields. Adding custom filter fields is a required setup step for building a YSR report.</p>
<i>See custom filter fields.</i>	
YSR Wizard	<p>Sequential layout of report setup screens introduced in Correspondence Plug-in 7.9. For information, see “The Wizard” on page 4.</p>

## CHAPTER 2

# YSR Report Setup

### In this chapter:

Report Setup .....	16
Report Design Considerations.....	23
Voyager Analytics Report Sections .....	41
Scripted Report Sections.....	48
YSR Scripting Conventions .....	51
YSR Deliverables.....	57

NEW LEASE PRICING HEALTH RULES
RENEWAL LEASE PRICING HEALTH RULES

Show Inactive Health Rules

Gardenview Terrace 1 Bedroom - 94 Units (144 Phased Units) ▾
Unit group section

Availability (Units)
Predicted 30 Day Availability (Units)
Leased (Units)
Occupancy (Units)
Predicted 30 Day Trend (Units)
Exposure (Units)
Vacancy Forecast (Units)
Vacant Unrented (Units)

5  
(16)
10  
(15)
89  
(128)
87  
(120)
84  
(129)
4  
(29)
18  
(47)
5  
(16)

Decrease Rent Recommendation > Availability, Leased
+ Add New Rule

Increase Rent Recommendation > Availability, Leased
+ Add New Rule

Turn Cost Allocation > Availability

This chapter describes how to set up the basic framework of a YSR report. Complete setup steps for all YSR report components are not provided in this section. This section is intended to provide an overview of setup steps, design considerations, and definitions of screen elements.



This section describes how to set up YSR reports manually. You can also design YSR reports with the Yardi Excel Add-In. For more information, see "Yardi Excel Add-In for YSR" on page 242.

# Report Setup

## In this section:

Adding YSR Reports.....	17
Copying and Moving YSR Reports .....	21
Managing Inactive YSR Reports.....	21
Finding and Displaying Missing and Inactive YSR Reports .....	22

To set up a YSR report, you must set up the three components of all YSR reports:

- Report templates (Excel, Word, and PDF files).
- Data sources (SQL scripts or Voyager analytics).
- The YSR report design (completed in Voyager).

The YSR report design includes the following:

- Definition of the YSR report filter (the custom filter fields that appear to users).
- Top-level select statement and key column.
- Filter mappings.
- Report section definitions.
- Email, attachment, and report output settings.

## Adding YSR Reports

The first step to creating a YSR report is to add a report code and description to your database. You can return later and add the YSR report filter (the custom filter fields available to users), add sub-reports and report sections, and specify the report templates included in the report.

After you add your YSR report, next steps include:

- Add one or more report sections (data sources) and report templates (data destinations).  
For information, see Chapter 4, "YSR Report Section and Template Setup."
- Define the YSR report filters (custom filter fields).  
For information, see "YSR Report Filter Definition" on page 62.
- Map filters.  
For information, see "Filter Mapping for Voyager Analytics Report Sections" on page 77 and "Filter Mapping for Scripted Report Sections" on page 81.
- Add a top-level select statement.  
For information, see "Top-Level Select Statements and Key Columns" on page 24.

### To add a YSR report

- 1 From the side menu, select **Admin > YSR Correspondence > Setup Report**. The **YSR Correspondence Setup Screen** filter appears.
- 2 Click **Submit**. The **YSR Report Setup** screen appears.
- 3 Click **New**. The fields on the screen become empty.
- 4 Complete the screen. For field descriptions, see "YSR Report Setup Screen Reference" on page 18.
- 5 Click **Save**.

## YSR Report Setup Screen Reference

The screenshot shows the YSR Report Setup screen. At the top, there are fields for 'Code' (CommCh), 'Description' (Comm Charges Demo), and 'Top-Level Select Statement' containing the query: 'select p.scode PropCode, p.saddr Name from property p where 1=1 #condition1#'. Below these are 'OR Script File' (empty) and 'Key Column' (PropCode). There is also an 'Inactive' checkbox and a 'Notes' area. A toolbar at the bottom includes Save, New, Save Copy, Define Filters, Map Top-Level Filter, Dump SQL, Delete Setup, and Help buttons. Below the toolbar is a navigation bar with tabs: Report Setup (selected), Attachment & Email, Output Options, and Digital Signature. At the very bottom is a grid table with columns: Order, Report Code, Template File, Script File, Page Break Column, Map Script Filter, Sections, Inactive?, and Delete?. One row is visible in the grid.

### Top-Level Select Statement

Determines how results appear on the report generation screen.

Optional if your report contains just one Excel-based report section; required in all other cases.

**TIP** The top-level select must retrieve a unique identifier for the reporting object (like a property hmy or scode). All data sources used in the YSR report must retrieve the same key column, and you must identify each data source's key column by its alias on the **Report Section Setup** screen.

For more information, see “Top-Level Select Statements and Key Columns” on page 24.

**NOTE** Do not surround your statement with //SELECT or //END SELECT. For example, you can enter a query like the following:

```
SELECT p.scode PropCode, p.sAddr1 PropName FROM Property p
WHERE 1=1 #Condition1#
```

In this example, PropCode can serve as the key column. If you use this example, make sure all other report sections also retrieve property scodes. Then, on the **Report Sections Setup** screen, specify the scode alias used by each report section in the **Key Column** field.

**NOTE** Enter either a select statement in this field or a SQL script in the **OR Script File** field. Do not enter information in both fields.

### OR Script File

The .txt file that contains the (unnamed) top-level select statement. Surround the unnamed select statement with //SELECT and //END SELECT. Store this file in your Reports path.

**NOTE** Complete this field or the **Top-Level Select Statement** field. Do not enter information to both fields.

**NOTE** Such an unnamed select statement must be the only unnamed statement in the .txt file. If any additional statements are present, they must be named.

### Key Column

Enter the alias of the data column, queried by the top-level select statement, that returns the key column for the YSR report. The top-level select statement must include this data column.

**TIP** The key column controls report iteration. For example, if you use property scodes or hmys, Voyager generates one report per property. If you use a tenant identifier, Voyager generates one report per tenant.

**NOTE** As a best practice, do not use ‘property’ as the alias for the key column. Key columns aliased ‘property’ can cause reports to display no data for users with property list security. This applies to all aliases for all key columns in the report.

<b>Inactive</b>	Deactivates the report. Users are not able to select the report from the <b>Report Name</b> field on the report generation screen, and the report setup screens are not available for editing.  <b>TIP</b> To reactivate the report, select <b>Admin &gt; YSR Correspondence &gt; Setup Report</b> . On the <b>YSR Correspondence Setup</b> screen, select <b>Yes</b> from the <b>Inactive?</b> drop-down list. Locate your report, clear the <b>Inactive</b> check box, and save the report.
<b>Exclude from Generate List</b>	Prevents the report from appearing in the drop-down list on the <b>Report Generation</b> screen. The report is still active, however, and is available for use in Task Runner, Report Scheduler, and from custom side menus.
<b>Define Filters</b>	Opens the <b>Report Filters Setup</b> screen where you can set up the YSR report filter. For more information, see “Defining YSR Report Filters” on page 65.
<b>Map Top-Level Filter</b>	Opens the <b>Map Top-Level Filter</b> screen where you can manage filtration for the top-level select statement. For more information, see “Filtering the Top-Level Select Statement” on page 86.
<b>Dump SQL</b>	Downloads a .pkg file containing all the elements of the YSR report, for use in an SQL database which is at the same plugin level as the source database. This feature is primarily designed to move a YSR report from one database, such as ‘Test’, to a receiving database, such as ‘Live’.
<b>Delete Setup</b>	Deletes all elements of the report design. Does not delete any supporting script files or report templates.
<b>Order</b>	The merge order of the sub-reports.
<b>Report Code</b>	A unique identifier of maximum sixteen characters for each sub-report. Earlier versions of YSR supported 8 characters only.  This code appears on the corresponding <b>Report Sections Setup</b> screen (accessible when you click the <b>Sections</b> button  ).
<b>Template File</b>	The Excel or Word document that contains text, formatting, and placeholders for Voyager data (smart markers in Excel; merge fields in Word). Save the report template to the Reports path.  YSR supports .xls, .xlsx, .doc, and .docx file formats.
<b>Script File</b>	The filename of the SQL script file Voyager uses to generate this sub-report (a .txt file). Save the template to Voyager’s reports path.  <b>NOTE</b> You can also use Voyager analytics, rather than a SQL script (or in addition to a SQL script), to retrieve data for your report. If you want to use Voyager analytics data sources only, leave this field blank.
<b>Page Break Column</b>	The alias of the column containing the values that Voyager uses to label the worksheets of merged Excel reports.  If you supply a page break column, Voyager uses this worksheet naming convention when rendering Excel reports: <ul style="list-style-type: none"><li>• If a template has named worksheets: WorksheetName-KeyColumnName</li><li>• If a template has unnamed worksheets: ReportCode-KeyColumnName</li></ul> Applicable to reports that use a top-level select statement and key column only.
<b>Map Script Filter</b> 	Access point for the <b>Map Script Filter</b> screen corresponding to the SQL script file supporting each sub-report.  For more information about filter mapping, see Chapter 3, “Filtering Data with YSR.”.

---

<b>Sections</b>	Access point for the <b>Report Sections Setup</b> screen. On this screen you can:
	<ul style="list-style-type: none"> <li>Specify the key column for each report section in a sub-report.</li> <li>Add section codes and map them to named SELECT sections or Voyager analytics data sources.</li> </ul> <p>For more information, see “Report Section Setup” on page 101.</p>
<b>Inactive</b>	Excludes the corresponding sub-report from the final report output.
<b>Delete</b>	Deletes the corresponding sub-report when saving changes to the screen.

---

### • Output Options Tab

The screenshot shows the 'Output Options' tab of the Report Setup interface. It contains four main sections: 'Screen Output Options', 'Scheduler Report Setup', 'QR Code Setup', and 'Report Generate Help'. Under 'Screen Output Options', there are three checkboxes: 'Enable New Screen Output' (checked), 'Enable Tabbed Screen Output' (unchecked), and 'Use Freeze-Pane Setting for Report Title/Header' (unchecked). Under 'Scheduler Report Setup', there are three checkboxes: 'Regular Report' (unchecked), 'Long Running Report' (unchecked), and a text input field for 'Long Running Script File Name'. Under 'QR Code Setup', there is a text input field for 'QR Data Source' with the placeholder '(Provide SectionCode.ColumnName)'. Under 'Report Generate Help', there is a text input field for 'Help Topic'.

---

<b>Enable New Screen Output</b>	Headers appear slightly differently on screen and you can display Excel-based tables and charts on screen.
<b>Enable Tabbed Screen Output</b>	(This check box becomes active when you select the <b>Enable New Screen Output</b> check box). For use with tabbed Excel report templates. Tabs appear on screen, each displaying separate worksheets from the YSR report workbook.
<b>Use Freeze-Pane Setting for Report Title/Header</b>	(This check box becomes active when you select the <b>Enable New Screen Output</b> check box). For use with Excel report templates. Applies bold formatting and increased font size to the first row of Excel template when generated to screen.
<b>Scheduler Report Setup</b>	Users can generate the report using Report Scheduler on an ad hoc basis, from the Report Generation screen. For more information about setting up this option, see “Generating Ad Hoc Reports in Report Scheduler” on page 137.
<b>QR Code Setup</b>	Applicable if your reports display QR codes. For more information about displaying QR codes, see “Generating QR Codes Dynamically at Runtime” on page 139.
<b>Help Topic</b>	Topic marker of web help article that opens when users click the <b>Help</b> button on the report generation screen.

---

## Copying and Moving YSR Reports

This section describes how to create copies of reports and how to move a report from one database to another.

You can create a copy of an existing report by opening the report, editing its report code, and clicking the **Save Copy** button.

Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?
10	RentRoll	yDoc_Property_rentroll.xlsx					<input type="checkbox"/>	<input type="checkbox"/>



If you copy a report, consider creating copies of all report template files and script files as well. Save the files with a new filename and edit the YSR report setup in Voyager so that the copy report refers to the new filenames. Otherwise both the original and the copy reports refer to the same files, and any edits to the report templates or script files affect both YSR reports.

### Moving YSR reports

You can move a YSR report from one database to another by clicking the **Dump SQL** button on the **YSR Report Setup** screen. Voyager creates a .pkg file containing all the elements of the YSR report (filter definitions, filter mappings, section setup, and so on), for use in an SQL database which is at the same plugin level as the source database. Download the file, save it, and use Voyager Workstation Administration to load the .pkg file into another database.

Make sure to save all report template files and script files to the Reports path in the receiving database as well.

## Managing Inactive YSR Reports

You can view a list of the inactive YSR reports in your database and modify the status of each report without visiting each setup screen.

### To manage inactive YSR reports

- From the **YSR Admin** side menu, select **Setup > YSR Administration**. The **YSR Administration** screen appears.



For information about the **YSR Admin** side menu, see "Adding the YSR Admin Menu" on page 162.

- Click the **Inactive YSR Reports** tab.

YSR Code	YSR Name	Inactive
GAB_KathyM	GAB_KathyM Progress Residential	<input checked="" type="checkbox"/>
InvoiceR	Invoice Register	<input checked="" type="checkbox"/>
WLFamilyHis	Waiting List Family History	<input checked="" type="checkbox"/>

- Clear the **Inactive** check box for each report that you want to make active.
- If you want to show all YSR reports, clear the **Show Only Inactive YSR Reports** check box.



When all YSR reports are shown, you can select the **Inactive** check box to deactivate multiple YSR reports at one time.

- Click **Save**.

## Finding and Displaying Missing and Inactive YSR Reports

You can find and display missing and inactive YSR reports.



YSR administrators should review menus and correct YSR report issues periodically so that users can avoid broken links and report generation failures.

### To find and display missing and inactive YSR reports

- From the **YSR Admin** side menu, select **Setup > YSR Administration**. The **YSR Administration** screen appears.



For information about adding the **YSR Admin** side menu, see "Adding the YSR Admin Menu" on page 162.

## 2 Click the **Menu Items with YSR Report Issues** tab.

All Menu Sets are included, except 'YSRAII'.

Menu Set	Menu Title	Link Caption	YSR Code	Status
iAffordable	Affordable Entire Set	Reports-Waiting List Reports-PHA Waiting List Reports-Family History	WLFamHis	Inactive
Inspections	Inspections	Reports-Inspection YSR Reports	InspLetter	Missing
iPHA	Housing Authority Manager	Reports-Waiting List Reports-Family History	WLFamilyis	Inactive
iPHA	Housing Authority Manager		WLFamilyis	Inactive
iWait		Public Housing Authority-Waiting Lists-Waiting List Review	WLReview	Missing
Performance	Performance Returns Manager	NCREIF-NCREIF Reports-NCREIF Composite YSR	NCREIF	Missing
Performance	Performance Returns Manager	GIPS-GIPS Reports-GIPS Composite YSR	GIPS	Missing
Performance	Performance Returns Manager		NCREIF	Missing
TIMenu	Tax and Insurance	Reports-Tax Reports- Tax Assessments	TIASMVAL	Missing
TIMenu	Tax and Insurance	Reports-Insurance Reports-Insurance Policies	TIINSPOL	Missing
TIMenu2	Tax Insurance 2	Reports-Tax Reports- Tax Assessments	TIASMVAL	Missing



Voyager reviews all the menu sets in your Voyager database, with the exception of the **YSRAII** menu set, and determines which of those menu items are used to generate YSR reports. Voyager then derives the YSR Report Code from the menu item link and checks it against the database to ensure that the YSR report is available and active. If the report is missing or inactive, it appears on the **Menu Items with YSR Report Issues** tab.

## Report Design Considerations

### In this section:

Top-Level Select Statements and Key Columns.....	24
Identifying Your Key Column .....	25
Formatting Columns on the Report Generation Screen.....	27
Working with Consolidated Versus Unconsolidated Data .....	30
Showing Both Consolidated and Unconsolidated Data .....	35
Properties, Property Lists, and Headers for Consolidated Reports .....	37
Formatting Columns on the Report Generation Screen.....	27
Working with Consolidated Versus Unconsolidated Data .....	30
Showing Both Consolidated and Unconsolidated Data .....	35
Properties, Property Lists, and Headers for Consolidated Reports .....	37
Read-Only Databases .....	40

This section discusses the role of the top-level select statement, the key column, and design considerations when working with consolidated data.



Topics related to consolidated data are intended for more experienced report writers. New report writers will benefit from the consolidation-related topics but may prefer to move to other sections of the documentation first.

## Top-Level Select Statements and Key Columns

The top-level select statement and the key column go hand-in-hand; whenever you use one, you must use the other. Conceptually, the key column is more important, for the following reasons:

- It controls report iteration. For example, if you use a property hmy or scode as a key column, Voyager creates one report per property. If you use tenant identifiers as the key column, Voyager creates one report per tenant.
- It enables Voyager to link data across multiple report sections. For example, suppose one report section retrieves property names and addresses, while another report section retrieves unit numbers. Voyager cannot link unit data to the correct properties unless the unit report section also retrieves a property identifier. The property identifier (whether an scode, hmy, or other identifier) is the key that coordinates the two report sections.

You cannot take advantage of the key column without also using a top-level select statement. The top-level select statement in itself, however, does not add to the body of the report. Its function is to organize results on the report generation screen.



The preferred key column for property-based reports is **property.hmy**, aliased **propertyid**.

### When to use a top-level select statement and key column

The top-level select statement and key column are mandatory if you want to:

- Include more than one sub-report.
- Attach reports to Voyager objects.
- Email reports to Voyager contacts or email IDs.
- Control how results appear on the report generation screen.
- Include Word report templates.
- Generate one report per key column value (rather than appending data to the same report)
- Merge Excel-based reports into one Excel document with one worksheet per key column value.

The top-level select statement and key column are optional if:

- The report includes just one sub-report.
- You want to publish the report immediately to screen.
- You want to generate just one Excel or PDF file.

## Top-level select statements and display of results

Whenever you use a top-level select, Voyager displays the top-level select columns, by alias, on the report generation screen. For example, if you use this top-level select statement:

```
SELECT p.scode PropertyCode, p.saddr1 PropertyName, p.hmy PropertyID
from property p where p.hmy in (#property#)
```



This example assumes that you have previously defined your filter, and included in that definition the named filter element “property” (hence the inclusion of #property# as a token in the above select statement). It also assumes that you have defined that element to be of the “ysiProperty” or similar type.

Voyager displays the following columns:

PropertyCode	PropertyName	PropertyID	Report
comoff01	Sunrise Tower	2.00	<a href="#">View Report</a>
cnret01	Otts Mall	39.00	<a href="#">View Report</a>

If you prefer, you can hide columns, apply different data formats, and edit column names by completing the top-level column setup process. For more information, see “Formatting Columns on the Report Generation Screen” on page 27.

## Identifying Your Key Column

The key column uniquely identifies the primary object of the YSR report. It controls report iteration, and it determines how Voyager links data from one report section to data from other report sections.

For example, suppose you have a YSR report that retrieves property and tenant data. Depending on your key column, you can generate one report per property or one report per tenant. In the great majority of cases, YSR reports generate one report per property. Therefore, the property hmy is the most commonly used key column. You can however use other unique identifiers like tenant IDs (hmyperson) if you want to establish a different breakdown of report data.

When selecting a key column, remember that every report section must retrieve a data column that contains the primary key. Otherwise, Voyager cannot relate data from one report section to data from another report section. This is not typically a limitation, since reports usually focus on a property, tenant, investor, unit, or similar, and you can associate all of these reporting objects to a common parent (to a property, typically) by their foreign keys. For example, the hprop column of the tenant table is a foreign key to the property table. It contains property hmys and links each tenant to a property, identified by property hmy.

In rare cases where you cannot link sub-reports by a common key, you cannot use a top-level select statement. For example, if you have one script that retrieves data from the PMUser table and one script that retrieves data from the Tenant table, you cannot link the results because the tables have no common foreign key. You must set up the PMUser report and the Tenant report as separate YSR reports.



The preferred primary key is the numerical identifier of your reporting object, like property.hmy.

You can in principle use string values, like property codes (property.scode), as the primary key. You must, however, use the numerical handle (the hmy) if you want to email or attach reports.

### Key columns for Voyager analytics data sources

If you are retrieving data from a Voyager analytics data source, you must identify the primary key used by the Analytics data source, and its alias. In the majority of cases, Voyager analytics use property hmys, aliased **propertyId**.

Exceptions include Investment Management data sources (investorid and investmentid) and Condo (ownerhmy).

You must identify the alias of the analytics data source's primary key on the **Report Section Setup** screen, as illustrated in the following graphic:

The screenshot shows the 'Report Sections Setup' dialog box. At the top, there are fields for 'YSR Report:' (Trial Balance (TrialBal)) and 'Report Code:' (TrialBal). Below these are 'Save', 'Close', and 'Help' buttons. The main area is titled 'Report Sections' and contains a table with one row. The table columns are: Section Code, Description, SELECT Name, Standard Analytics Report, Map Standard Analytics Filter, Tokens, Relations, Key Columns, Primary?, and Multiple Rows?. The row data is: TrialBal, Trial Balance, , Financial Analytics, , , , propertyId, checked, checked.

Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	Primary?	Multiple Rows?
TrialBal	Trial Balance		Financial Analytics				propertyId	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

### Key column aliases

You must identify the key columns used in each of your report sections by providing their aliases on YSR report setup screens.



As a best practice, do not use 'property' as the alias for the key column. Key columns aliased 'property' can cause reports to display no data for users with property list security. This applies to all aliases for all key columns in the report.

The following graphic shows how you might use multiple aliases for the same primary key. Aliases support modular report design principles.

#### Alias of key column as retrieved by the top-level SQL select

The screenshot illustrates the configuration of a report named 'Commercial Rent Roll' (RentRM). In the main setup window, the 'Top-Level Select Statement' field contains the query:

```
select p.hmy PropCode, p.saddr1 propname from property p
where 1=1 and p.hmy in (#propertylist#)
```

The 'Key Column' is set to 'PropCode'. The 'Notes' panel provides details about the report's working environment and filters.

A modal dialog titled 'Report Sections Setup' is open, showing the configuration for the 'Commercial Rent Roll (RentRM)' report. It lists two sections: 'RentRoll' and 'FutureL', both mapped to 'Commercial Analytics'.

Below the main setup window, a table titled 'Report Sections' lists the sections and their properties. An arrow points from the 'Sections' column of this table to the 'Sections' column in the 'Report Sections Setup' dialog, indicating the relationship between the two.

Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections
10	RentRM	yDoc_Property_rentrollM.xlsx				<input checked="" type="checkbox"/>
						<input checked="" type="checkbox"/>

Alias of key column used in sub-report sections

## Formatting Columns on the Report Generation Screen

When you add a top-level select statement to a report and generate the report in Excel, Word, or PDF, Voyager displays the data queried by the select statement in multiple columns on the report generation screen, followed by a **View Report** column with a link to each report.

You can customize column display as follows. You can:

- Hide columns
- Apply data formats
- Create drill-down links

## Drill-down links

If you want to create drill-down links for the data in any of the columns, make sure that your top-level select statement retrieves the numerical identifier of the object type of the drill-down links. For example, if you want to link to the **Property** screen, your top-level select statement must retrieve the hmy column of the property table. If you want to link to the **Owner** screen, you must retrieve the hmyperson column of the owner table. You will need these identity columns to create the drill-down links. See the procedures that follow for an example.



Top-level select statements have the same filter mapping requirements as other custom SQL scripts used in the report. For more information about filter mapping, see “Filtering the Top-Level Select Statement” on page 86.

## To organize results on the report generation screen

- 1 Select **Admin > YSR Correspondence > Setup Report**. The **YSR Correspondence Setup** screen appears.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see “Adding the YSR Admin Menu” on page 162.

- 2 Click **Submit**. The **YSR Report Setup** screen appears.

- 3 Add a top-level select statement and complete the **Key Column** field.



Add the top-level select statement directly on this screen or save it in a text file and enter the filename of the text file in the **Or Script File** field.

For purposes of illustration, this example uses the following top-level select statement (WHERE clause not shown):

```
select p.hmy PropertyID, p.saddr1 Name, o.ocode Owner, o.hmyperson OwnerCode from property p inner join propown po on po.HPROPERTY = p.hmy inner join owner o on po.howner = o.HMYPERSON
```

- 4 To hide columns, apply formatting, or create drill-down links, click the **Top-Level Column Setup** button. The **Top-Level Filter Setup** screen appears.

**Top-Level Filter Setup**

YSR Report: Commercial Rent Roll (RentRM)

**Save** **Close**

Column Name	Drill	Object Type	Drill URL	ID Column	Hide	Data Type
PropertyID	<input type="checkbox"/>	<input type="button" value="▼"/>			<input type="checkbox"/>	Integer <input type="button" value="▼"/>
Name	<input checked="" type="checkbox"/>	Property <input type="button" value="▼"/>	javascript:DrillDown('CommonProperty.aspx?&PropertyId=' +	PropertyID	<input type="checkbox"/>	<input type="button" value="▼"/>
Owner	<input checked="" type="checkbox"/>	Owner <input type="button" value="▼"/>	javascript:DrillDown('CommonOwner.aspx?OwnerId=' +	OwnerCode	<input type="checkbox"/>	<input type="button" value="▼"/>
OwnerCode	<input type="checkbox"/>	<input type="button" value="▼"/>			<input checked="" type="checkbox"/>	<input type="button" value="▼"/>
					<input type="checkbox"/>	<input type="button" value="▼"/>

- 5 Complete the fields. For more information, see “Top-Level Filter Setup Screen Reference” on page 30.
- 6 Click **Save**.
- 7 Click **Close**.
- 8 Generate your report using any output type other than **Screen**. Voyager displays the results of your top-level select statement (with customizations if applicable).

**Commercial Rent Roll**

Property	comoff01^cnoff01	Report Name	Commercial Rent Roll (RentRM) <input type="button" value="▼"/>													
FromDate	12/01/2008 <input type="button" value="▼"/>	Output Type	Excel <input type="button" value="▼"/>													
ReportType	RentRoll <input type="button" value="▼"/>	Merge Reports	<input type="checkbox"/>													
		Show Grid	<input type="checkbox"/>													
			Attach Reports <input type="checkbox"/>													
			Email Reports <input type="checkbox"/>													
			Publish To SharePoint <input type="checkbox"/>													
			Show on Portal <input type="checkbox"/>													
		<b>Generate</b>	<b>Clear</b>	<b>Help</b>												
<table border="1"> <thead> <tr> <th>PropertyID</th> <th>Name</th> <th>Owner</th> <th>Report</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>Sunrise Tower</td> <td>usowner</td> <td><a href="#">View Report</a></td> </tr> <tr> <td>108</td> <td>Airport Plaza</td> <td>ontent</td> <td><a href="#">View Report</a></td> </tr> </tbody> </table>		PropertyID	Name	Owner	Report	2	Sunrise Tower	usowner	<a href="#">View Report</a>	108	Airport Plaza	ontent	<a href="#">View Report</a>			
PropertyID	Name	Owner	Report													
2	Sunrise Tower	usowner	<a href="#">View Report</a>													
108	Airport Plaza	ontent	<a href="#">View Report</a>													
		Name and Owner columns have links			OwnerCode column is hidden											
					PropertyID column formatted as integer											

## Top-Level Filter Setup Screen Reference

The screenshot shows the 'Top-Level Filter Setup' dialog box. At the top, it displays 'YSR Report: Commercial Rent Roll (RentRM)'. Below this are 'Save' and 'Close' buttons. The main area is a grid table with the following columns:

Column Name	Drill	Object Type	Drill URL	ID Column	Hide	Data Type
PropertyID	<input type="checkbox"/>	<input type="button" value="▼"/>			<input type="checkbox"/>	Integer <input type="button" value="▼"/>
Name	<input checked="" type="checkbox"/>	Property <input type="button" value="▼"/>	javascript:DrillDown('CommonProperty.aspx?&PropertyId=' + this.value);	PropertyID	<input type="checkbox"/>	<input type="button" value="▼"/>
Owner	<input checked="" type="checkbox"/>	Owner <input type="button" value="▼"/>	javascript:DrillDown('CommonOwner.aspx?OwnerId=' + this.value);	OwnerCode	<input type="checkbox"/>	<input type="button" value="▼"/>
OwnerCode	<input type="checkbox"/>	<input type="button" value="▼"/>			<input checked="" type="checkbox"/>	<input type="button" value="▼"/>
	<input type="checkbox"/>	<input type="button" value="▼"/>			<input type="checkbox"/>	<input type="button" value="▼"/>

---

<b>Column Name</b>	The column in your top-level select statement to which customization applies. The text entered here must match exactly the field aliases used in the top-level select statement.
<b>Drill</b>	The column contains drill-down links. When selected, the <b>Object Type</b> and <b>Drill URL</b> fields become active.
<b>Object Type</b>	(Applicable when you select the <b>Drill</b> check box.)  The object type of the Voyager screen to which the drill-down link redirects users. Determines the string that appears in the <b>Drill URL</b> column.
<b>Drill URL</b>	(Applicable when you select the <b>Drill</b> check box.)  Voyager generates the string in this column automatically when you complete the <b>Object Type</b> field.
<b>ID Column</b>	(Applicable when you select the <b>Drill</b> check box.)  The field alias of the column in your top-level select that contains the unique identifier for the object relevant to the drill-down link.
<b>Data Type</b>	Voyager formats results using the data type selected in this field.

---

## Working with Consolidated Versus Unconsolidated Data

Consolidated reports aggregate data for multiple properties (or other report objects), while unconsolidated reports split out data per property (or other object). You can produce both types of reports in YSR, and you can also design reports that let users choose whether to consolidate.

Giving users the option to consolidate does present the report designer with some challenges. Designing consolidated reports in YSR can be challenging because of the role of the top-level select statement. The top-level select is essential for many report designs, but it also splits data into key column values rather than consolidating data. To generate consolidated data in a report with a top-level select, you must carefully consider the requirements of your underlying data source and write a top-level select that, if the user consolidates, hands off the appropriate primary key to the underlying data source.



To consolidate Financial Analytics-based report data, pass **1** as the primary key.

To consolidate Custom Financial Analytics-based report data, pass **-1** as the primary key.

See the following discussion for more information, example top-level select statements, and other options for script-based reports.

For purposes of illustration, suppose you are designing a report with only a property filter field and a check box for consolidation, as shown in the following graphic.

The two filter fields are defined as illustrated here:

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?
Property	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		<input type="checkbox"/>
IsConsolidate?	Consolidate?	2	Checkbox		<input checked="" type="checkbox"/>		<input type="checkbox"/>

The IsConsolidate check box resolves to '1' if true (if the user chooses to consolidate) and '0' if false (for unconsolidated reports). Use a CASE statement in your top-level select statement to handle each option. Examples follow. The examples are for illustration only and must be adapted to suit the filter fields in your report.

### This section includes the following procedures:

To offer consolidation for Financial Analytics-based reports.....	32
To offer consolidation for Custom Financial Analytics-based reports .....	32
To offer consolidation for script-based reports .....	33

#### To offer consolidation for Financial Analytics-based reports

Use property hmvs (p.hmy, aliased PropertyID below) as the key column for all report sections. Use the following top-level select statement to handle the user's choice about whether to consolidate:

```
//SELECT
SELECT distinct
PropertyCode = CASE #IsConsolidate# WHEN 1 then 'Consolidated' ELSE
p.scode END
,PropertyName = CASE #IsConsolidate# WHEN 1 then 'Consolidated' ELSE
p.saddr1 END
,PropertyID = CASE #IsConsolidate# WHEN 1 then 1 ELSE p.hmy END
FROM Property p
WHERE 1=1
AND p.hmy in (SELECT hproperty FROM listprop2 WHERE hproplist in
(#Property#))
AND itype = 3
//END SELECT
```



This script passes '1' as the key column when the user chooses to consolidate.

#### To offer consolidation for Custom Financial Analytics-based reports

Use property hmvs (p.hmy, aliased PropertyID below) as the key column for all report sections. Use the following top-level select statement to handle the user's choice about whether to consolidate:

```
//SELECT
SELECT distinct
PropertyCode = CASE #IsConsolidate# WHEN 1 then 'Consolidated' ELSE
p.scode END
,PropertyName = CASE #IsConsolidate# WHEN 1 then 'Consolidated' ELSE
p.saddr1 END
,PropertyID = CASE #IsConsolidate# WHEN 1 then -1 ELSE p.hmy END
FROM Property p
WHERE 1=1
AND p.hmy in (SELECT hproperty FROM listprop2 WHERE hproplist in
(#Property#))
AND itype = 3
//END SELECT
```



This script passes '-1' as the key column when the user chooses to consolidate.

### To offer consolidation for script-based reports

If you are working with a custom script, you can use either of the top-level select statements shown above or some adaptation of them. Most importantly, you must adapt the main scripted report section or sections to respond to the consolidation option passed in by the top-level select.

The following text provides an example top-level select statement and corresponding main report section for a report that retrieves occupied square footage from an ASCAP table.



Scripting is not recommended if you are working with financial data. Use Financial Analytics or Custom Financial Analytics instead to avoid introducing discrepancies between your YSR report and Voyager Financial Analytics reports.

```
//SELECT
SELECT distinct PropertyCode = CASE #IsConsolidate# WHEN 1 then 'Consolidated' ELSE p.scode END
,PropertyName = CASE #IsConsolidate# WHEN 1 then 'Consolidated' ELSE p.saddr1 END
,PropertyID = CASE #IsConsolidate# WHEN 1 then -1 ELSE p.hmy END
FROM Property p
WHERE 1=1
AND p.hmy in (SELECT hproperty FROM listprop2 WHERE hproplist in
(#Property#))
AND itype = 3

//END SELECT

//SELECT MAIN
SELECT
CASE WHEN UPPER('#IsConsolidate#') = '1' THEN -1 ELSE p.hmy END
[PropertyID]
, ddate [kpdate]
, sum(dococcupiedarea) [OccupiedAreaSum]
FROM ASCAP_COMMKPI k
INNER JOIN Property p on p.hmy = k.hProp
WHERE 1=1
and p.hmy in (#Property#)

GROUP BY CASE WHEN UPPER('#IsConsolidate#') = '1' THEN -1 ELSE p.hmy END, dtDate

//END SELECT
```

The preceding example script uses some simplifications for purposes of clarity. If you copy and paste the example select statements into a report for testing, you will notice some differences in sort order when you generate the report for consolidated data, as compared to unconsolidated.

One way to fix the sort order is shown below.

```
//SELECT Main
DECLARE @IsConsolidate int
SET @IsConsolidate = (#IsConsolidate#)

IF (@IsConsolidate = 0)
BEGIN

    SELECT
        CASE WHEN @IsConsolidate = '1' THEN -1 ELSE p.hmy END [PropertyID]
        , dtdate [kpdate]
        , sum(dococcupiedarea) [OccupiedAreaSum]
    FROM ASCAP_COMMKPI k
    INNER JOIN Property p on p.hmy = k.hProp
    WHERE 1=1
    and p.hmy in (#Property#)

    GROUP BY CASE WHEN @IsConsolidate = '1' THEN -1 ELSE p.hmy END, dtDate,
    p.HMY

END

ELSE IF (@IsConsolidate = 1)
BEGIN

    SELECT
        CASE WHEN @IsConsolidate = '1' THEN -1 ELSE p.hmy END [PropertyID]
        , dtdate [kpdate]
        , sum(dococcupiedarea) [OccupiedAreaSum]
    FROM ASCAP_COMMKPI k
    INNER JOIN Property p on p.hmy = k.hProp
    WHERE 1=1
    and p.hmy in (#Property#)
```

```

GROUP BY CASE WHEN @IsConsolidate = '1' THEN -1 ELSE p.hmy END, dtDate
END

//END SELECT

```

## Showing Both Consolidated and Unconsolidated Data

In some cases, users may want to view both consolidated and unconsolidated data in the same report. In most cases, you must use a workaround to accommodate this request.

### Workarounds

If you want to combine consolidated and unconsolidated data in the same report, consider the following alternative approaches:

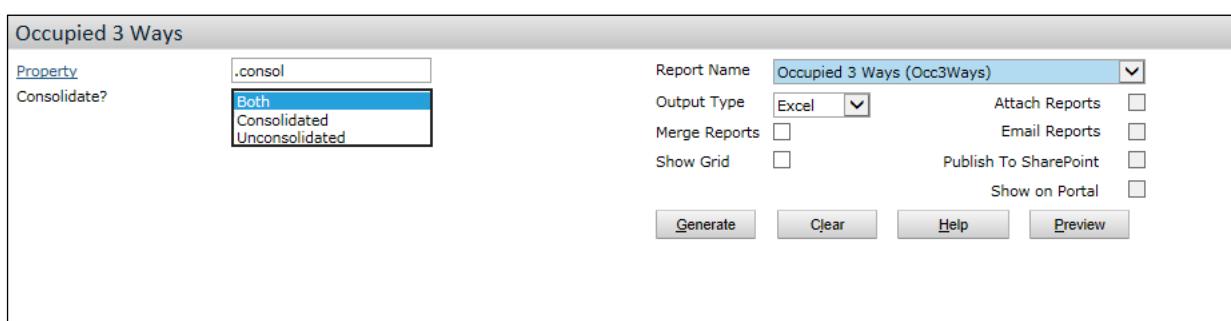
- Use Report Scheduler to generate two YSR reports separately and combine them in one report packet.
- Use YSR to generate two YSR reports (one consolidated, one unconsolidated) as XLSMs. Include in the XLSM a macro that copies the worksheet in the consolidated report to the unconsolidated workbook. Save the combined file as an XLSX and distribute.
- Generate the consolidated report and save it as an attachment in the Voyager reports path. Then add the rendered report as a static attachment to the unconsolidated report and generate.

### Scripted solutions

With scripted reports, you may be able to give users the option to generate a report for unconsolidated data, consolidated, or both. The example report illustrated here uses property scodes for the key column. Applicability of this approach is therefore limited, since Financial Analytics and Custom Financial Analytics require the property hmy as the key column instead.

This example report retrieves property square footage from an ASCAP table. Adapt it for other scripted reports that retrieve data from ASCAP or other custom tables.

The example report, 'Occupied 3 Ways', has the following filter fields:



The **Consolidate** filter field is configured as a list with three possible string values, 'Consolidated,' 'Unconsolidated,' and 'Both.'

Report Filters Setup								
YSR Report: Occupied 3 Ways (Occ3Ways)								
<input type="button" value="Save"/> <input type="button" value="Preview"/> <input type="button" value="Help"/>								
								
Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	List Values
Property	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
IsConsolidate	Consolidate?	2	List		<input type="checkbox"/>		<input type="checkbox"/>	Consolidated^Unconsolidated

The top-level select and the main select statements are as follows:

```
//SELECT
Select
--For Consolidated Data
'Consolidated' [PropCode]
from property p
where l=1
and p.hmy in (#Property#)
and '#IsConsolidate#' in ('Consolidated', 'Both')

UNION

Select
--For Unconsolidated Data
p.sCode [PropCode]
from property p
where l=1
and p.hmy in (#Property#)
and '#IsConsolidate#' in ('Unconsolidated', 'Both')

//END SELECT

//SELECT MAIN

SELECT
CASE WHEN UPPER('#IsConsolidate#') in ('Consolidated', 'Both') THEN
'Consolidated' ELSE p.scode END [PropCode]
, dtdate [kpdate]
, sum(dococcupiedarea) [OccupiedAreaSum]
```

```

FROM ASCAP_COMMKPI k
INNER JOIN Property p on p.hmy = k.hProp
WHERE l=1
and p.hmy in (#Property#)

GROUP BY CASE WHEN UPPER('#IsConsolidate#') in ('Consolidated', 'Both')
THEN 'Consolidated' ELSE p.scode END, dtDate

UNION ALL

SELECT
    p.scode [PropCode]
, dtdate [kpodate]
, sum(doccupiedarea) [OccupiedAreaSum]
FROM ASCAP_COMMKPI k
INNER JOIN Property p on p.hmy = k.hProp
WHERE l=1
and p.hmy in (#Property#)
and '#IsConsolidate#' in ('Both')

GROUP BY p.scode, dtDate

//END SELECT

```

## Properties, Property Lists, and Headers for Consolidated Reports

Reports that feature consolidated data usually use the `ysiPropertyOrListLookup` for the top-level property filter field. This lookup list has the advantage of accepting single properties (`comoff01`), multiple caret-separated properties (`comoff01^cnoff01`), and property lists (`.consol`). This list has the disadvantage, however, that it immediately converts any property list code into its component property codes. This presents difficulties if you intend to display the property list code anywhere in the rendered report (most likely, in the header of the report).

This section illustrates one way to display either the property list or the component property codes in the report header, depending on the user's choice whether or not to consolidate. This example is applicable to Excel-based reports only.



For background information about property lists and another approach to working with lists, see "YSI Property Lookup Lists" on page 62.

To create a report template that displays either a property list or individual properties:

- In the top-level YSR report filter, include a property filter field (using ysiPropertyOrListLookup) and a check box for consolidation (see fields 1 and 6 in this graphic).

**Report Filters Setup**

YSR Report: Budget Comp with Dynamic Header for Consolidated vs Unconsolidated

**Save** **Preview** **Help**

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?
PropertyCode	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Book	Book	2	Lookup List	ysiBookLookup	<input type="checkbox"/>		<input type="checkbox"/>
TreeCode	Tree	3	Lookup List	ysiAccountTreeLookup	<input type="checkbox"/>	1	<input type="checkbox"/>
FromPeriod	From Period	4	Post Month		<input type="checkbox"/>		<input checked="" type="checkbox"/>
ToPeriod	To Period	5	Post Month		<input type="checkbox"/>		<input checked="" type="checkbox"/>
IsConsolidate	Consolidate	6	Checkbox		<input type="checkbox"/>		<input type="checkbox"/>
SupressZero	Supress Zero	7	Checkbox		<input type="checkbox"/>		<input type="checkbox"/>
					<input type="checkbox"/>		<input type="checkbox"/>



Do not select the **Show in Title** check box (not shown, but visible on the right side of the screen) for the property field.

- Add a scripted report section for retrieving header data. This script must retrieve property hmys and scodes when #IsConsolidate# is true (resolves to '1'), and it must retrieve nothing when #IsConsolidate# is false (resolves to '0').

```
//SELECT HEADER
DECLARE @IsConsolidate int
SET @IsConsolidate = (#IsConsolidate#)

select p.hmy propertyid,
p.scode propertycode,
p.saddr1 as propertyname
from property p
where p.hmy in (
select hproperty from listprop2 where hPropList in
(#PropertyName#) and itype in (select case when @IsConsolidate <> 0
then 11 else 3 end))

//END SELECT
```

- In a hidden worksheet in your Excel report template, add:

- A smart marker that stands in for the property codes retrieved by the scripted header report section.

- A smart marker that stands in for the literal string the user supplies in the property filter field at run time (a property list, most likely). To do this, use the built-in section code YSRFilterValues and the field name of your property filter field.

A	B	C	D
1 &=Header.PropertyName	&=YSRFilterValues.PropertyCode		
2			
3			
4			

Sheet1 Sheet2 (+)



If the user chooses to run the report consolidated, no data appears in column A. The header select section is designed to retrieve data only when the report is run unconsolidated.

- 4 In the report template header rows, use conditional logic to display (or hide) the property list, depending on whether or not anything appears in column A of the hidden worksheet (the column containing individual property codes *if*, and only if, the user generates the report unconsolidated.)

E4	:	X	✓	fx	=IF(Sheet2!A1="",Sheet2!B1,"")
A				E	F
1					
2	<b>Budget Comparison</b>				
3					
4	Property List:				
5	Property :	&=Header.propertyname			
6	&=YSRFilterValues.ReportTitleValues				
7					

Cell E4 is configured so that, if the column containing individual property codes is empty, then it displays the property list column; otherwise it displays nothing (a blank string, "").

- 5 Use similar conditional logic or conditional formatting to show and hide the headers in column A. You do not need to apply conditional logic around the smart marker for the individual property codes (&=Header.propertyname); the Header report section is scripted so that they will not appear if the user consolidates.

The rendered report appears as illustrated below:

## Consolidated

Budget Comp with Dynamic Header for Consolidated vs Unconsolidated Data

Property	.actprop
Book	
Tree	
From Period	01/2007
To Period	12/2008
Consolidate	<input checked="" type="checkbox"/>

Report Name	Budget Comp with Dynamic Header for Consoli
Output Type	Excel
Merge Reports	<input type="checkbox"/>
Show Grid	<input type="checkbox"/>
Generate	Clear
Help	Preview

A E F

1

2 **Budget Comparison**

3

4 Property List: .actprop

5

6 From Period: 01/2007 To Period: 12/2008

7

8

9 Account Code Account Description PTD Actual

10

## Unconsolidated

Budget Comp with Dynamic Header for Consolidated vs Unconsolidated Data

Property	.actprop
Book	
Tree	
From Period	01/2007
To Period	12/2008
Consolidate	<input type="checkbox"/>

Report Name	Budget Comp with Dynamic Header for Consoli
Output Type	Excel
Merge Reports	<input type="checkbox"/>
Show Grid	<input type="checkbox"/>
Generate	Clear
Help	Preview

A E

1

2 **Budget Comparison**

3

4

5 Property : Sunrise Tower

6 From Period: 01/2007 To Period: 12/2008

7

8

9 Account Code Account Description

10

## Read-Only Databases

If you want to generate a report from a read-only database, insert the following lines before any select statements:

```
//OPTIONS
WITH_READONLY
//END OPTIONS
```

## Voyager Analytics Report Sections

### In this section:

Adding Voyager Analytics Report Sections.....	42
Adding Extra Columns of Data to Voyager Analytics Report Sections.....	45

Voyager analytics data sources are powerful tools for YSR reports. When you use Voyager analytics data sources in YSR, you can be sure your data will match Voyager data exactly because you are using built-in Voyager tools. You can also set up Voyager analytics sources in YSR without writing any custom SQL scripts. YSR setup screens refer to Voyager analytics as standard analytics reports.

While you can generate Voyager analytics reports without YSR, there are benefits to using them in YSR. With YSR, you can:

- Add and subtract columns of data to standard Voyager analytics. (Applicable to some analytics only; for information, see “Adding Extra Columns of Data to Voyager Analytics Report Sections” on page 45).
- Modify the formatting of your reports by manipulating your templates in Word or Excel.
- Create custom filters to alter the options available to users when generating reports.

To set up YSR to generate an analytics report, you must have an Excel or Word template ready to use with your YSR report. You can:

- Create a new template using the Yardi Excel Add-In (recommended).

For more information, see Chapter 8, “Yardi Excel Add-In for YSR.”

- Edit an existing Voyager YSR analytics report template.

Standard installations of YSR include a limited number of out-of-the-box YSR reports and report templates. Voyager stores these templates in the Reports path.

For a quick start, identify an out-of-the-box template that matches the analytics report you want to generate. For example, to set up YSR to generate a Trial Balance report, you can use the Excel template in your Reports path titled YSR\_TrialBalance.xlsx.

You can download the template, edit it, and save it to the Reports path with a new filename so that you do not overwrite the existing template.



When modifying out-of-the-box YSR reports, rename all supporting files (templates and scripts) and reintroduce the YSR report to Voyager with a new report code. This ensures that, when you update your Plugins, your modified report is not overwritten.

- Create a new template. Use the Yardi Excel Add-in to create smart markers in an Excel template.

- If you want to write smart markers manually, see the section titled “Identifying Field Names for Voyager Analytics Reports” on page 108. This section explains how to discover the field aliases (column names) used in many standard Voyager analytics. You must know the field aliases in order to set up your report templates.

## Adding Voyager Analytics Report Sections

This section provides basic procedural steps for adding a Voyager analytics report section. It is not exhaustive, and the procedures contain several references to fuller discussions of topics covered elsewhere in the documentation.

### Basics of adding analytics report sections

To set up a report section to retrieve data using Voyager analytics, you must complete the following basic tasks:

- Add to the YSR report filter the filter fields appropriate to the Voyager analytics you are working with. For more context, see “Defining Filter Fields for Use with Voyager Analytics” on page 69.
- At the report section level, identify the analytics you want to use by selecting it in the **Standard Analytics Report** column.
- At the report section level, map your filters. For an extended discussion, see “Filter Mapping for Voyager Analytics Report Sections” on page 77.

### Availability of Voyager analytics

If the Voyager analytics you require does not appear in the list of available analytics on the **Report Sections Setup** screen, verify that the .PKG file that exposes the analytics to YSR has been executed against your database.



You can find a list of all available .PKG files in the Yardi Excel Add-In on the **Analytic Data Sources** tab of the **Yardi Spreadsheet Reporting Setup** screen.

When you execute one of the .PKG files, you add one or more records into the CustomCorrespStandardList table. The sName column of the CustomCorrespStandardList table contains the values that appear in the **Standard Analytics Report** drop-down list.

Typically, the list of available analytics increases with the automatic execution of .PKG files during the installation of updated plug-ins. As development advances, the .PKG files associated with each plug-in introduce new interfaces to standard Voyager analytics reports.

## To add a Voyager analytics report section

- Select **Admin > YSR Correspondence > Setup Report**. The **YSR Correspondence Setup** screen appears.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see “Adding the YSR Admin Menu” on page 162.

- Click **Submit**. The **YSR Report Setup** screen appears.

The screenshot shows the **YSR Report Setup** interface. In the top left, the **Code** field is set to **RentRoll**, and the **Description** field is set to **Commercial Rent Roll**. The **Top-Level Select Statement** field contains the SQL query: `Select p.scode PropertyCode, p.saddr1 Name from property p where 1=1 #condition1#`. The **Key Column** is set to **PropertyCode**. On the right, the **Notes** pane displays the configuration details: `ReportType : RentRoll  
MainQuery : Yes  
IsMarketRent : Yes  
SummaryType : Property  
Filters : PropertyList,FromDate`. Below the main form are buttons for **Save**, **New**, **Save Copy**, **Define Filters**, **Map Top-Level Filter**, **Dump SQL**, **Delete Setup**, and **Help**. At the bottom, tabs for **Report Setup**, **Attachment & Email**, **Output Options**, and **Digital Signature** are visible. A table below lists report sections, with one entry for **RentRoll** (Order 10, Template File: **yDoc\_Property\_rentroll.xlsx**, Script File: blank, Page Break Column: blank, Map Script Filter: checked, Sections: checked, Inactive?: unchecked, Delete?: unchecked).

- Complete the fields in the top section of the screen. (The top-level select statement is optional if your report has just one report section.)
- Complete the fields on the **Report Setup** tab.

The **Script File** field is not necessary if you are setting up YSR to generate a report using Voyager analytics only. Use it if your report also contains a scripted report section.



You can also ignore the **Map Script Filter** button on the **Report Setup** tab. You must navigate to the **Report Sections Setup** screen to find the **Map Standard Analytics Filter** button that applies to Voyager analytics sub-reports.

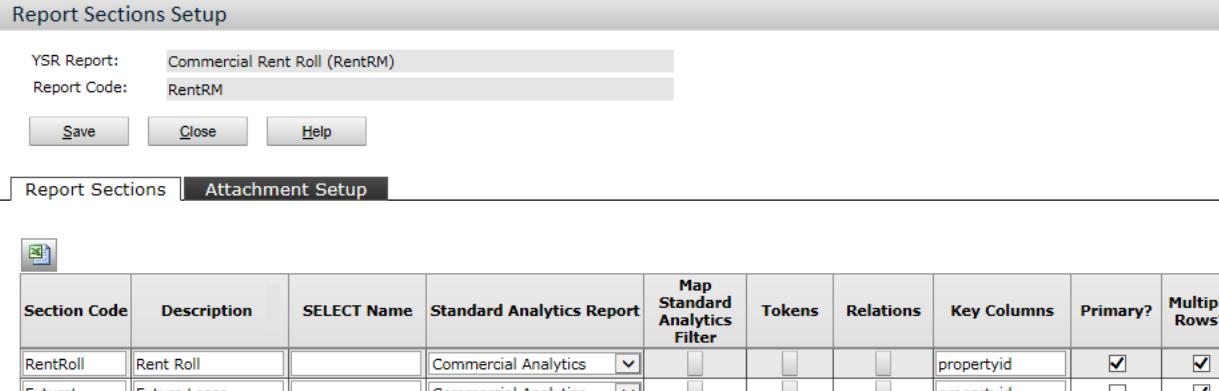
- Define the top-level YSR report filter. Click **Define Filters** and add the filter fields that are appropriate to your analytics data source.



In a subsequent step you will map these filter fields to the filter elements native to your analytics data source. For more discussion of which fields are necessary, see “Defining Filter Fields for Use with Voyager Analytics” on page 69.

- Click **Save**.

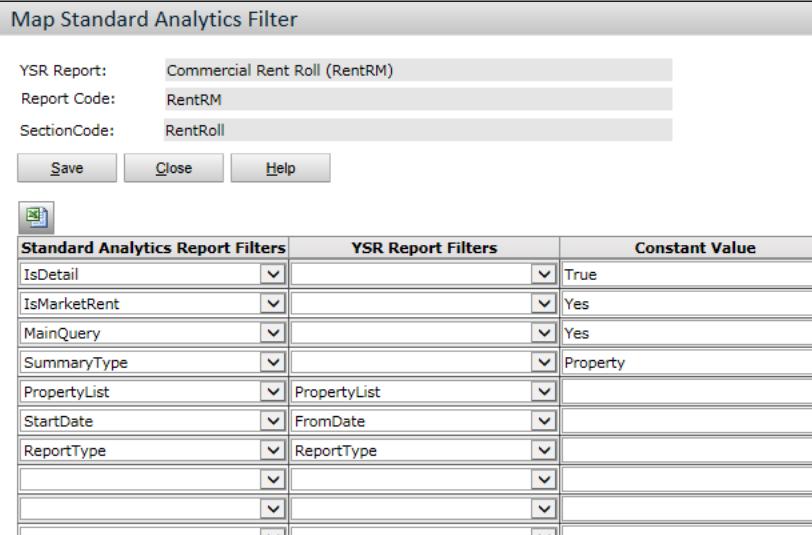
- 7 On the **Report Setup** tab, click the **Sections** button  corresponding to your sub-report. The **Report Sections Setup** screen appears.



The screenshot shows the "Report Sections Setup" dialog box. At the top, it displays the YSR Report as "Commercial Rent Roll (RentRM)" and the Report Code as "RentRM". Below this are three buttons: "Save", "Close", and "Help". The main area is titled "Report Sections" and contains a table with two rows:

Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	Primary?	Multiple Rows?
RentRoll	Rent Roll		Commercial Analytics				propertyid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
FutureL	Future Lease		Commercial Analytics				propertyid	<input type="checkbox"/>	<input checked="" type="checkbox"/>

- 8 Complete the fields. For more information, see “Report Sections Setup Screen Reference” on page 103.
- 9 Click the **Map Standard Analytics Filter** button  in the row corresponding to the analytics report. The **Map Standard Analytics Filter** screen appears.



The screenshot shows the "Map Standard Analytics Filter" dialog box. It displays the YSR Report as "Commercial Rent Roll (RentRM)", the Report Code as "RentRM", and the SectionCode as "RentRoll". Below these are three buttons: "Save", "Close", and "Help". The main area contains a table with columns: "Standard Analytics Report Filters", "YSR Report Filters", and "Constant Value". The table has 10 rows, each mapping a standard analytics filter to a YSR report filter. The first few rows are as follows:

Standard Analytics Report Filters	YSR Report Filters	Constant Value
IsDetail		True
IsMarketRent		Yes
MainQuery		Yes
SummaryType		Property
PropertyList		PropertyList
StartDate		FromDate
ReportType		ReportType
		
		

- 10 Map your YSR report filter fields to the standard analytics report filters.



You do not need to map every option in the **Standard Analytics Report Filters** section. Complete the filter mapping for only the standard analytics filters that your report needs. It may be helpful to compare the filter of the underlying analytics when completing filter mapping.

For a full discussion of analytics filter mapping, see “Filter Mapping for Voyager Analytics Report Sections” on page 77.

- 11 Complete the **Constant Value** column as necessary. (Use it to hard-code elements of the Voyager analytics filter.) Known values for the **Constant Value** column are listed in Appendix D, “Constant Values for Analytics Data Source Filter Mapping.”

12 Click **Save**.

13 Click **Close**.

## Adding Extra Columns of Data to Voyager Analytics Report Sections

You can use YSR to add a column of data to some Voyager analytics reports. You can add additional data columns to Voyager analytics data sources that return a string data type (as opposed to a .NET data table).

There are a few different ways to identify the analytics data sources that can be extended with extra columns. You can:

- Click the **Tokens** button on the **Report Sections Setup** screen after selecting an analytics data source from the **Standard Analytics Report** drop-down list. If the **Custom Tokens** screen appears, you can add an extra column.
- Use the Yardi Excel Add-in. Review the list of analytics on the **Standard Analytics Data Sources** tab of the **Yardi Spreadsheet Reporting** screen. Analytics that can be extended have an asterisk (\*) next to their name.
- Review the sReturnType column of the CustomCorrespStandardList table. If the analytics returns a SQL query, the sReturnType column contains 'string'.

### To add an extra column of data to Voyager analytics reports

- 1 Select **Admin > YSR Correspondence > Setup Report**. The **YSR Correspondence Setup** screen appears.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see "Adding the YSR Admin Menu" on page 162.

**2 Click Submit.** The **YSR Report Setup** screen appears.

The screenshot shows the 'YSR Report Setup' interface. At the top, there are fields for 'Code' (RentRoll), 'Description' (Commercial Rent Roll), 'Top-Level' (Select p.scode PropertyCode, p.saddr1 Name from property), 'Select Statement' (p where 1=1 #condition1#), 'Key Column' (PropertyCode), and 'Inactive' (checkbox). Below these are buttons for 'Save', 'New', 'Save Copy', 'Define Filters', 'Map Top-Level Filter', 'Dump SQL', 'Delete Setup', and 'Help'. A 'Notes' panel on the right contains the following text:

```
ReportType : RentRoll
MainQuery : Yes
IsMarketRent : Yes
SummaryType : Property
Filters : PropertyList, FromDate
```

At the bottom, there are tabs for 'Report Setup' (selected), 'Attachment & Email', 'Output Options', and 'Digital Signature'. Below the tabs is a table for managing report sections:

Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?
10	RentRoll	yDoc_Property_rentroll.xlsx						

**3 On the Report Setup tab, click the Sections button** corresponding to your sub-report. The **Report Sections Setup** screen appears.

The screenshot shows the 'Report Sections Setup' interface. At the top, it displays 'YSR Report: Commercial Rent Roll (RentRoll)' and 'Report Code: RentRoll'. Below are buttons for 'Save', 'Close', and 'Help'. At the bottom, there are tabs for 'Report Sections' (selected) and 'Attachment Setup'.

Below the tabs is a table for managing report sections:

Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	Primary?	Multiple Rows?
RentRoll	RentRoll		Commercial Analytics				propertyid	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
FutureL	Future Lease		Commercial Analytics				propertyid	<input type="checkbox"/>	<input checked="" type="checkbox"/>
								<input type="checkbox"/>	<input checked="" type="checkbox"/>

- 4 Click the **Tokens** button ■ corresponding to the section you want to extend. The **Custom Token Setup** screen appears.

Token Name	Select Query	Inactive?	Delete?
PropCust	(SELECT sAcquire FROM Property WHERE sCode=PropCode)	<input type="checkbox"/>	<input type="checkbox"/>
UnitCust	(SELECT PUCode FROM Unit WHERE sCode=UnitCode AND hProperty=PropertyID)	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>



If the **Tokens** button is not active, the corresponding Voyager analytics data source does not support the addition of custom tokens (it does not return a string data type).

- 5 Complete the fields.

<b>Token Name</b>	Specify a unique name to use as the column alias.
<b>Select Query</b>	<p>Enter a SELECT query, enclosed in parentheses, to identify the data column you want to add. The SELECT query should retrieve one data column only, and you must write the WHERE clause sufficient to establish a 1-to-1 association with the individual data records being rendered by the Analytics data source.</p> <p>The WHERE clause may require just one identity (as in the PropCust example above) or multiple (as in the UnitCust example). For each identity, the left-hand side (sCode, for example) is an unaliased field name from the collection of fields exposed by the FROM clause. The right-hand side (PropCode, for example) is the aliased name of a field returned by the Analytics data source.</p> <p>Identifying the aliases used by an Analytics data source can be difficult. The Yardi Excel Add-In provides a list of fields by Analytics data source in the Smart Marker Designer. You can also trigger a SQL error by entering junk data in the <b>Custom Token Setup</b> screen. When you generate the report, Voyager displays the SQL behind the whole YSR report. Use a formatter to read the SQL and identify the aliased field names. For more information about this method, see "To identify the field aliases used in Voyager analytics" on page 109.</p>
<b>Inactive</b>	Excludes the custom data from the analytics report.

- 6 Click **Save**.

- 7 Click **Close**.

## Scripted Report Sections

This section describes how to add scripted report sections to a YSR report. A scripted report section uses a custom SQL script to retrieve data.

### Adding Scripted Report Sections

You can add one or more scripted report sections to any YSR report. If your report contains more than one report section (of any type, Voyager analytics or scripted), you must add a top-level select statement to your report to coordinate the sub-reports.

When you use a script, there are two ways to manage filtration. You can:

- Use named tokens (#FilterFieldName#) in your script.
- Use numbered tokens (#conditionN#) in your script and add corresponding filter clauses (Val 1 parameters) in the Voyager YSR user interface on the **Map Script Filter** screens.

If you use (#FilterFieldName#), you can directly access the value returned by the corresponding custom filter field in the YSR report filter. For example, a script that contains the following:

```
WHERE p.hmy in (#property#)
```

resolves to:

```
WHERE p.hmy in (2)
```

at run time.

On the other hand, if you use numbered conditions (#conditionN#) and add the Val 1 parameter on the **Map Script Filter** screen, you can more easily re-use your script in different YSR reports with different filter field naming conventions or mapping requirements.

For more information and detailed examples, see “Filter Mapping for Scripted Report Sections” on page 81.

#### To add a scripted report section

- 1 Select **Admin > YSR Correspondence > Setup Report**. The **YSR Correspondence Setup** screen appears.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see “Adding the YSR Admin Menu” on page 162.

- 2 Click Submit.** The **YSR Report Setup** screen appears.

The screenshot shows the 'YSR Report Setup' interface. At the top, there are fields for 'Code' (set to 'CommCh'), 'Description' (set to 'Comm Charges Demo'), 'Top-Level Select Statement' (containing a SQL query), and 'Key Column' (set to 'PropCode'). Below these are buttons for 'Save', 'New', 'Save Copy', 'Define Filters', 'Map Top-Level Filter', 'Dump SQL', 'Delete Setup', and 'Help'. A 'Notes' area is also present. The bottom section features tabs for 'Report Setup' (selected), 'Attachment & Email', 'Output Options', and 'Digital Signature'. Below the tabs is a table with columns: Order, Report Code, Template File, Script File, Page Break Column, Map Script Filter, Sections, Inactive?, and Delete?. One row is visible in the table, showing values for Order 1, Report Code commch, Template File CommRent.xlsx, Script File CommRent.txt, Page Break Column Id, and Map Script Filter.

- 3 On the Report Setup tab, complete the Order, Report Code, and Template File fields.**

<b>Order</b>	The display order of the reports included in the YSR report.  <b>TIP</b> You can leave space between numbers in case you want to add reports to the report packet later.
<b>Report Code</b>	A code for organizing reports.
<b>Template File</b>	The template containing the text and formatting for your report.  For more information about templates, see "YSR Report Setup" on page 15.
<b>Script File</b>	The text file containing the SQL script you want to use to retrieve data.

- 4 Click Save.**

- 5 On the **Report Setup** tab, click the **Map Script Filter** button corresponding to your sub-report. The **Map Script Filter** screen appears.

YSR Report Filters	Script Filter Condition	
property (#Condition1#)	p.hmy in (#property#)	
tenant (#Condition2#)	t.hmyperson in (#tenant#)	
date (#Condition3#)	ckd.dtdate in ('#date#')	

- 6 Complete the fields. For information, see “Filter Mapping for Scripted Report Sections” on page 81.
- 7 Click **Save**.
- 8 Click **Close**.
- 9 On the **Report Setup** tab, click the **Sections** button corresponding to your sub-report. The **Report Sections Setup** screen appears.

Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	Primary?	Multiple Rows?
ch	charges	charges	<input type="button" value="▼"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	property	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
img	images	images	<input type="button" value="▼"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	property	<input type="checkbox"/>	<input checked="" type="checkbox"/>



You must use the values that appear in the **Section Code** column in the smart markers for the corresponding report template. Likewise, your SQL script must use the section names that appear in the **SELECT Name** column. For example, suppose the charges select section returns a field aliased ChargeDate. To display ChargeDate data in your report output, use this smart market in an Excel report template: &=ch.ChargeDate

- 10 Complete the fields. For information, see “Report Sections Setup Screen Reference” on page 103.
- 11 Click **Save**.
- 12 Click **Close**.

# YSR Scripting Conventions

## In this section:

Select Statements, Named and Unnamed.....	51
Session and User-Related Tokens .....	52
DDLs, DMLs, and Temporary Tables .....	54
Example Scripts.....	56

This section describes YSR scripting conventions.

YSR script files do not need to follow all Yardi SQL scripting conventions. They require only //SELECT...//END SELECT statements.

When authoring a YSR script, it is best to include a //NOTES...//END NOTES section where you describe the purpose of the script. Provide detailed notes on the filter elements and any unusual subtleties of the select statement.

## Select Statements, Named and Unnamed

YSR scripts can have one or more SELECT statements. If your script has multiple //SELECT statements, leave the top-level //SELECT section unnamed and provide unique names for every other //SELECT section. Voyager interprets the unnamed //SELECT as the top-level select statement for the report. Voyager interprets named //SELECT sections as the scripts for sub-reports. Map the names of your //SELECT sections to the section codes used in your report templates by completing the **Report Sections Setup** screen.



The first line of a named SELECT section must follow strict syntax requirements. It must use the following (case-insensitive) syntax:

//SELECT SectionName

- Insert exactly one space between SELECT and the section name.
- Do not include anything else, including comments, in this line.
- The section name can include up to 100 characters.

Voyager ignores any sections that do not meet these requirements. Similar requirements apply for the //END SELECT line (excepting the section name; add the section name to the first //SELECT only).

## Session and User-Related Tokens

To manage data related to session and user, you can use session-related tokens in your YSR scripts. The following table lists supported tokens and example return values. For an example of a script that uses tokens, see “Example Script File With a Single Select Statement” on page 56.

Token	Description	Example value
#@@ATTACHREPORTS#	Returns True if the user selects the <b>Attach Reports</b> check box on the report generation screen; False otherwise.	True
#@@ACCTCODE#	Applies account masking to acct.scode. Useful for GL account reports. If you select acct.scode, you get unformatted account numbers (say, 11100000). If you select #@@ACCTCODE# instead, you get formatted accounts (say, 1110-0000). Your script must retrieve data from the ACCT table, aliased AS <b>a</b> .  <b>TIP</b> #@@ACCTCODE# is not a string data type. Do not enclose in single quotes in your script.	1110-0000
#@@ACCTCODE^TabAlias#	Same as #@@ACCTCODE#, but accepts alternate aliases for the ACCT table.  For example, if you alias the ACCT table AS <b>b</b> , use #@@ACCTCODE^b#.	1110-0000
#@@DigitalSignature#	Placeholder for a click-able digital signature in a rendered report. Requires additional setup. For more information, see “Setting up Digital Signatures” on page 144.	
#@@EMAILREPORTS#	Returns True if the user selects the <b>Email Reports</b> check box on the report generation screen; False otherwise.	True
#@@GUID#	Returns the global unique identifier of the temporary file associated with the YSR report.	6682dfd0-a5b7-4f73-9189-7bddb01fedd1
#@@MERGEREPORTS#	Returns True if the user selects the <b>Merge Reports</b> check box on the report generation screen; False otherwise.	True
#@@OUTPUTTYPE#	Returns the user’s selection from the <b>Output Type</b> field on the report generation screen.	EXCEL

#@@PUBLISHTOSHAREPOINT#	Returns True if the user selects the <b>Publish to SharePoint</b> check box on the report generation screen; False otherwise.	True
#@@SESSIONID#	A unique number that accompanies the browser login session key. Voyager executes multiple YSR SELECT sections in parallel; each returns the same session ID.	4555
#@@SHOWGRID#	Returns True if the user selects the <b>Show Grid</b> check box on the report generation screen; False otherwise.	True
#@@USERLANGUAGEID#	Returns the value of PMUser.hLanguage for secured users. Returns 0 for the DBO.	6
#@@USERCOUNTRYID#	Returns the value of Country_Info.hMy for secured users. Returns 0 for the DBO (international only).	12
#@@USERID#	Returns the value of PMUser.hMy for secured users. Returns 0 for the DBO.	14
#@@USERNAME#	Returns the value of PMUser.uName for secured users. Returns blank for the DBO.	Alex
#@@WEBSHARENAME#	Returns the URL of the active webshare. Potentially useful for building drill-down links.	<a href="http://yardidemo/demo708pi7">http://yardidemo/demo708pi7</a>

## DDLs, DMLs, and Temporary Tables

As of Correspondence Plug-in 7, YSR supports scripted reports that use temporary tables.

### DDLs and DMLs

In some cases you may need to process data definition language (DDL) or data manipulation language (DML) commands before or after executing a select statement.

To ensure that Voyager processes any DDLs and DMLs in the proper order (before or after the main select statement), use the following conventions:

**//Select No Crystal** Voyager processes any DDLs or DMLs contained in the //Select No Crystal... //End Select section before executing the report.

**//Select No Crystal After** Voyager processes any DDLs or DMLs contained in the //Select No Crystal After... //End Select section after executing the report.

The following graphic shows an example script with DDLs and DMLs before and after the main select statement.

```

YSR_NoCrystal.txt - Notepad
File Edit Format View Help
//Select No Crystal
IF NOT EXISTS (SELECT * FROM sys.tables WHERE object_id = OBJECT_ID
(N'[dbo].[temp_cryst]'))
create table temp_cryst
(
tenantcode varchar(100),
tslastname varchar(100),
username varchar(100)
)

insert temp_cryst
select sCode,slastname,'#@@USERNAME#'
from Tenant
//end select

//select test_no_crystal
select tenantcode sCode ,tslastname sLastname from temp_cryst
//end select

//Select No Crystal After
delete from temp_cryst where username = '#@@USERNAME#'
//End select

```

Select section containing DDL and DML to process before the main select

Main select statement

Select section containing DML to process after the main select

### Temporary Tables

If you need to create a local temporary table and subsequently retrieve data from that table for consumption by YSR, you can:

- Create and use the table within a single named SELECT section (recommended). The table is dropped when the session goes out of scope or when your named SELECT explicitly drops it.

- Create a table in a //SELECT NO CRYSTAL section. The table remains in scope and you can use it in other named SELECT sections.



If you use this method, you may encounter database errors if multiple sessions run concurrently. If you schedule your reports with Task Runner, stagger your reports so that they do not execute simultaneously.

To create local temporary tables, use the SQL Server convention of starting an object name with the number sign (#).

The following text provides an example of a script that uses a temporary table (#PROPDETAILS) in multiple SELECT sections.

```
//SELECT NO CRYSTAL
Select * into #PROPDETAILS from Property p where l=1 #condition1#
select hmy from #PROPDETAILS
//END SELECT

//SELECT GetProperty
Select hmy, scode code, saddr1 addr from #PROPDETAILS P
//END SELECT

//SELECT GetUnit
Select hmy, scode code from Unit U
where l=1
and hproperty in (select hmy from #PROPDETAILS)
#condition2#
//END SELECT

//SELECT GETBUILDING
Select hmy, sbldgid code from building where hprop in (select hmy from
#PROPDETAILS P)
#condition3#
//END SELECT
```

## Before and After Hooks

If you want to run SQL commands before or after YSR generates a report, use the `YSR_Generate_After_[report]` and `YSR_Generate_After_[report]` hooks, respectively.

For example, to run a procedure after generating report\_1, begin with the following line:

```
CREATE PROCEDURE YSR_Generate_After_report_1
```

## Example Scripts

This section provides an example of a YSR script with a single, unnamed select statement (suitable for a top-level select statement only), and a YSR script with multiple, named select statements suitable as data sources for sections of a sub-report.

### Example Script File With a Single Select Statement

```
//SELECT
Select
    pr.uCode                      "Owner Code"
    ,pr.uLastName                  "Owner Name"
    ,pr.hMy                        OwnerID
    ,pr.sAddr1                     Address
    ,pr.sCity                      City
    ,Convert(Date, '#Date1#')      AsOfDate
    ,'#@@USERNAME#'                UserName
    ,(Select sDesc From Country_Info Where hMy = #@@USERCOUNTRYID#)
UserCountry
FROM
    Person pr
    Inner Join Property p on p.hLegalEntity=pr.hMy

WHERE pr.iPersonType=2
    #Condition01#
    #condition02#

Group By
    pr.hMy
    ,pr.uCode
    ,pr.uLastName
    ,pr.sAddr1
    ,pr.sCity

//END SELECT
```

### Example Script File With Two Named Select Sections

```
//SELECT Images
Select
p.scode                         property
,p.saddr1                        name
,Concat ('\\ServerName\Reports\Pictures\',ltrim(rtrim(p.scode)),'.jpg')
```

```

AS PictureName
FROM property p
WHERE l=1 #condition1#
//END SELECT

//SELECT Charges
Select distinct
    p.hmy
    ,p.scode
    ,t.slastname
    ,ckd.hunit
    ,ckd.dtdate
    ,ckd.dLeaseMonthlyRent
    ,ckd.dLeaseMonthlyCam
    ,ckd.hmy
    prophmy
    property
    tenant
    unit
    date
    MonthlyRent
    MonthlyCAM
    ckdhmy

from CommKPIDetail ckd
inner join property p on ckd.hprop = p.hmy
inner join tenant t on ckd.htenant = t.hmyperson

where l=1 #conditions#

//END SELECT

```

## YSR Deliverables

Network administrators or report writers who are writing YSR reports for clients, or moving a YSR report from one database to another, must prepare the following deliverables:

- A Yardi Package (.pkg) file for the setup.

A YSR .pkg file contains all the structural elements of a YSR report, including complete definition of custom filter fields, filter mapping, report sections, templates, and specification of data sources.

There are multiple ways to create a PKG file. You can:

- Set up a YSR report in Voyager and then click **Dump SQL** on the **YSR Report Setup** screen.
- Use the Yardi Excel Add-In and its Dump SQL function. For more information, see “Building YSR Reports with the Add-In (Dump SQL)” on page 267.
- Create a PKG file manually using any text editor (difficult and not recommended).

For a representation of the schema tables manipulated in the PKG file, see Appendix A, "YSR Report Schema."



If you load a PKG into a database that already contains a YSR report with the same YSR report code, Voyager overwrites the existing YSR report and all its components (custom filter fields, filter mapping, report sections, templates, and data sources).

There is no warning and there is no option to undo an inadvertent PKG upload. If you accidentally overwrite a YSR report, you must recreate and restore an equivalent PKG that describes the same report code.

- An Excel or Word template for every sub-report in the YSR report.
- All Yardi script files required to generate the YSR report.
- All supporting packages as necessary if, for example, your report uses a Custom Financial Analytics template or custom account trees.

### To set up YSR deliverables

- 1 Load the YSR setup package and any other customization packages through Voyager System Administration. For more information, see the *Voyager Core Administration and Setup Guide*.
- 2 Upload all necessary script files and Excel or Word templates to the Reports path.



You can upload files to the Reports path from Voyager System Administration: **Admin > Correspondence Configuration > Correspondence Document Manager**. Use the **Browse** and **Upload** buttons at the bottom of the screen.

You can also upload files to the reports path from the **YSR Admin** side menu: **Setup > YSR Administration**. Use the **Browse** and **Upload** buttons on the **Upload YSR Template or Script** tab. For information about the **YSR Admin** side menu, see "Adding the YSR Admin Menu" on page 162.

- 3 Create a menu entry for the YSR report. For more information, see "Side Menus" on page 158.



This step is optional. Users can also generate the report by clicking **YSR Correspondence > Generate Report** and selecting the report from the **Report Name** field.

## Troubleshooting

### In this section:

Error Messages .....	59
Other Problems.....	59

This section provides tips for common problems and error messages.

## Error Messages

### Column “not found in table....

You forgot to define the key column for one or more report sections.

### Cannot convert value to date: [CellName]

Check the data-type formatting in your Excel template. Row 1 must have **General** data type. Apply date-type and other formatting to rows 2 and following.

### Screen report must have an Excel template

This error can occur if you try to generate a report without an Excel-based template to screen. If your report does have an Excel template and you see this error:

- Check for typos in the filename of the template on the **Report Setup** tab (for example, Myreporttemplate.lxs).
- Check to see if you accidentally entered your template filename in the script column and your script, ending in .txt., in the report template column on the **Report Setup** tab.

## Other Problems

### Report runs to Excel and PDF but not to screen

You may have data-type problems in your Excel report template. Make sure that all cells in the header row, or row 1, have **General** date type. For an illustration, see “Smart Markers and Excel Data Types” on page 203.

### Report with multiple worksheets does not display on screen

Turn on **New Screen Output** and **Tabbed Screen Output** on the **Output Options** tab.

Alternatively, make sure that each worksheet has the same number of columns. If your report design requires a worksheet with a smaller number of columns, add a blank space or two in any cell in the column equivalent to the last column of your largest worksheet.

### Your script works fine in SQL Server Management Studio, but retrieves no data in YSR.

Check the //SELECT.... //END SELECT lines in your script file. Typos and additional characters, including blank spaces, can cause the select section to fail.

Check to make sure your section codes are mapped properly and your template contains the right section codes.

## CHAPTER 3

# Filtering Data with YSR

### In this chapter:

Filtration Overview .....	60
YSR Report Filter Definition .....	62
Filter Mapping for Voyager Analytics Report Sections .....	77
Filter Mapping for Scripted Report Sections.....	81

This section explains how to filter data with YSR.

## Filtration Overview

Every YSR report has at least two filter elements: the custom filter fields the user completes at run-time (the YSR report filter), and the filter requirements of the underlying data-retrieval engine (a SQL script or Voyager analytics). To filter data, therefore, you must co-ordinate each filter element in the report by *mapping filters*. Mapping filters refers to the process of linking filter elements in one part of a YSR report to elements in another part.

### Advantages of Filter Mapping

The advantage of YSR's filter mapping apparatus is that it provides maximum flexibility in report design. Filter mapping supports:

- Several types of filter technologies (YSI lookup lists, custom lookup lists, and more).
- Several types of token substitution (named tokens, numbered tokens, and session-related tokens).
- Both custom SQL scripts and Voyager analytics.
- Both hard-coded filters and filters that accept the user's input at run-time.
- Modular report design (re-use of elements of one report in another report).

The filter-mapping process differs depending on whether you are using custom SQL scripts or Voyager analytics to retrieve data for your report.

If you are using custom SQL scripts, the mapping process is more transparent in that it involves writing WHERE clauses directly in your script or on filter mapping screens.

If you are using Voyager analytics, the mapping process is more managed. You do not have to write any WHERE clauses, but you must be familiar with the filter elements native to the Voyager analytics engine. When you map filters, you either assign a constant value to the analytics report filters or map them to the custom YSR filter fields the user completes at run-time.

## Classic Versus New Filter View

With Correspondence Plug-in 7.5, a new report generation screen helps you take advantage of limited screen space. With collapsible filter sections and support for up to 3 columns, the new report generation screen provides a streamlined view for users on mobile platforms.

The following graphics show the example AR Aging Summary YSR report in the new view:

### Filters collapsed

This screenshot shows the 'AR Aging Summary Report' interface with the 'Report Options' section expanded. Under 'Report Options', the 'Report' dropdown is set to 'AgingSum' and the 'Output' dropdown is set to 'PDF'. To the right of these dropdowns are checkboxes for 'Merge', 'Grid', 'Attach', 'Email', 'Publish To SharePoint', and 'Show on Portal'. Below these options are three buttons: 'Generate', 'Clear', and 'Help'. Below the 'Report Options' section is a collapsed 'Report Filters' section indicated by a plus sign (+).

### Filters expanded (single column)

This screenshot shows the same 'AR Aging Summary Report' interface, but the 'Report Filters' section is now expanded. The expanded section contains several filter fields: 'Owner/Entity' (with a dropdown menu showing 'nlfund'), 'Property' (with a dropdown menu showing 'nlfund'), 'As of Month (mm/yyyy)' (with a dropdown menu showing '12/2017'), 'Tenant' (empty), 'Account' (empty), and 'Charge' (empty). The rest of the interface, including the 'Report Options' section, remains the same as in the collapsed view.

The new view is available at this location: [../pagesCustomCorrespGenerateNew.aspx](#)

To adopt the new view, add the new report generation page as a custom side menu. For information about setting up custom side menus, see "Setting Up Side Menus" in the *Core Administration and Setup Guide*.

# YSR Report Filter Definition

## In this section:

YSI Property Lookup Lists .....	62
Defining YSR Report Filters .....	65
Defining Filter Fields for Data Range.....	70
Defining Dynamic Lists and Custom Lookups .....	72
Displaying User's Filter Criteria in Title of Report (Dynamic Titles) .....	75

This section describes how to create a YSR report filter (the top-level filter that users complete when they generate the report).

Every YSR report requires custom filter fields. These are the filter fields that appear to users at run-time, and which make up the YSR report filter. They differ by report.

Most YSR reports require, at a minimum, a field for filtering by property. Not only are properties the basic reporting objects in Voyager databases, but by using a property filter field, you apply property-level security to your YSR report. Users can generate reports only for those properties to which they have access. By default property filter fields on YSR reports display the property or property list selected on the user's dashboard.

There are a variety of ways to add a property filter field. A common way to add a property filter field is to leverage the `ysiPropertyOrListLookup` list. For more information about YSI property lookup lists, see "YSI Property Lookup Lists" on page 62.

In addition to a property filter field, you must add the filter fields that are important for your report design. You might need a field for filtering by account tree, or a field for filtering by G/L book type. On the other hand, you might hard-code account trees and G/L books into your report design. One way to ease the report design and filter mapping process is to start with one custom filter field only and add filter fields as necessary after your report is working.

## YSI Property Lookup Lists

The most common way to add a custom filter field for properties is to use one of the standard YSI property lookup lists. You can also create your own custom property list or lookup list.

Standard YSI property lookup lists include the following:

- `ysiPropertyOrListLookup`
- `ysiPropertyListLookup`
- `ysiPropertyLookup`
- `ysiPropertyOrDCFPropertyLookup`

If a user filters for one or more properties, all of the property lookup lists behave in the same way: they return the hmys of each property.

If a user filters for a property list, however, the property lookup lists behave differently. The following table summarizes the differences:

YSR Lookup List	Basic behavior	Behavior with respect to property list
<code>ysiPropertyOrListLookup</code>	Returns properties or property lists.	Returns the hmy of any individual properties in the selection as well as the component property hmays of any selected property lists, parsed out of those property lists.
<code>ysiPropertyListLookup</code>	Returns properties or property lists.	Returns the hmy of the property list only. Does not return the hmays of properties in the list.
<code>ysiPropertyLookup</code>	Returns properties only.	Does not accept property lists as filter criteria. Returns hmy of individual properties.
<code>ysiPropertyOrDCFPropertyLookup</code>	For ABF Valuations reports. Returns properties or model properties.	Returns the hmays of individual properties.



Property lists appear in the Property table with their own unique property hmy and with `itype = 11`. Real properties have `itype = 3` and budgeting and forecasting properties have `itype = 443`.

The ListProp2 table is a simple data construct maintained by Voyager for the purpose of determining which property lists contain which properties.

## Working with Both Properties and Property Lists

If you are designing a YSR report that supports filtration by property and by property list, the most obvious choice of lookup is the `ysiPropertyOrListLookup`. This list has one significant constraint, however: when the user filters by a property list, Voyager parses the property list into the hmays of its component properties. You cannot recover the hmy of the property list itself for use later in a sub-report.

For example, suppose you are designing an account tree report with a header that states the code of the property list in the header of the report.

sCode of property list					
	A	B	C	D	E
1	Trial Balance				
2	Books = Accrual				
3	Property = .comprop AND mm/yy = 01/15 - 12/15				
4					
5	Entity	Account	Description	Balance Forward	Debit
6	cnoff01	1110-0000	Cash - Operating	\$1,520,799.00	\$10,000.00
7	cnoff01	1310-0000	Receivable - Tenants	\$214,702.00	
8	cnoff01	2010-0000	Accounts Payable - Trade	\$0.00	
9	cnoff01	2051-0000	Sales Tax Payable - Output	-\$199,489.00	
10	cnoff01	2052-0000	Sales Tax Reclaimed - Input	\$40,797.00	
11	cnoff01	2175-0000	Security Deposit	-\$108,000.00	
12					

sCode of one property within a property list

If, in this example report, you use the `ysiPropertyOrListLookup` list, you cannot recover the sCode of the property list used to generate the report (.comprop).

If you instead use `ysiPropertyListLookup`, you can recover the sCode of the property list (.comprop), but you cannot retrieve any data because Voyager does not store G/L data on property lists.

The solution to this dilemma is to use `ysiPropertyListLookup` and edit your filter mappings. The `ysiPropertyListLookup` returns the hmy of the property list only, but you can use your filter mappings to expand the property list into its component properties.



Filter mappings contain the Val 1 parameters that Voyager appends to the WHERE clause in the SQL script used to retrieve data for the report. A common Val 1 parameter for a property lookup is `p.hmy in (#Property#)`, where `Property` is the field name of the YSR property filter field. For more information, see “Filter Mapping for Scripted Report Sections” on page 81.

Edit the Val 1 parameter so that Voyager retrieves both the hmy of the property list and the hmys of its component properties. For example, use the following syntax:

- `p.hmy in (select hproperty from listprop2 where hPropList in (#Property#))`

**Map Script Filter**

YSR Report:	Comm Charges Demo (CommCh)
Report Code:	commch
<b>Script Filter Condition</b>	
YSR Report Filters	Script Filter Condition
property (#Condition1#)	<code>p.hmy in (select hproperty from listprop2 where</code>

**Save**   **Close**   **Help**

With the edited Val 1 parameter, you can use `ysiPropertyListLookup` to return both the hmy of a property list and the hmys of its component properties.

For an alternative approach, useful when you want to give the user the option to consolidate data, see “Properties, Property Lists, and Headers for Consolidated Reports” on page 37.

## Defining YSR Report Filters

Every YSR report requires a report filter. The report filter is the set of custom filter fields that appear to users at run-time.

When you set up a report filter, you can use a set of standard Yardi filtering tools like YSI lookup lists, date and post month filters, and binary check boxes. You can also build your own custom filter fields using SQL or simple caret-separated lists (Yes^No^Maybe).

### Parent-Child Relationships

When you add custom filter fields, you can institute parent-child relationships so that Voyager restricts the values in one field based on the user’s selection in another field. For example, you can make a tenant lookup list the child to a parent property list. Voyager displays only the tenants in the property the user selects.

Other common examples include:

- Batch number and Control number

Add a number-type field for batch numbers and a lookup list for control numbers. Assign the batch number field as the parent to the control number field. Voyager shows only the control numbers in the batch the user selects.

- Date and Budget (ABF Valuations)

Add a date-type field and a lookup list for budgets. Assign the date field as the parent to the budget field. Voyager shows only the budgets related to the date the user selects.

### To define a YSR report filter

- 1 Select **Admin > YSR Correspondence > Setup Report**. The **YSR Correspondence Setup** screen appears.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see “Adding the YSR Admin Menu” on page 162.

- 2** Complete the filter and click **Submit**. The **YSR Report Setup** screen appears.

The screenshot shows the 'YSR Report Setup' interface. At the top, there are fields for 'Code' (set to 'CommCh'), 'Description' (set to 'Comm Charges Demo'), and a 'Top-Level Select Statement' containing SQL code. Below these are buttons for 'Save', 'New', 'Save Copy', 'Define Filters', 'Map Top-Level Filter', 'Dump SQL', 'Delete Setup', and 'Help'. A 'Notes' area is also present. The bottom section contains tabs for 'Report Setup', 'Attachment & Email', 'Output Options', and 'Digital Signature'. A large table below lists report definitions with columns for Order, Report Code, Template File, Script File, Page Break Column, Map Script Filter, Sections, Inactive?, and Delete?.

Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?
1	commch	CommRent.xlsx	CommRent.txt	Id				

- 3** Click **Define Filters**. The **Report Filters Setup** screen appears.

The screenshot shows the 'Report Filters Setup' interface. It includes a 'YSR Report:' dropdown set to 'Pictures Matching Rev R Pix (CodeToMatchPI10)', and buttons for 'Save', 'Preview', and 'Help'. Below is a table of filter fields with columns for Field Name, Label, Sequence, Type, Lookup Name, Multi Select?, Parent, Mandatory?, List Values, Code to ID, Query?, Column View, Show In Title, Inactive?, When Populated In URL, and Delete?.

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	List Values	Code to ID	Query?	Column View	Show In Title	Inactive?	When Populated In URL	Delete?
Property	Property	1	Lookup List	ysiPropertyLookup	<input type="checkbox"/>		<input checked="" type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tenant	Tenant	2	Lookup List	ysiTenantLookup	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DateFrom	Date From	3	Post Month		<input type="checkbox"/>		<input checked="" type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DateTo	Date To	4	Post Month		<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- 4** Complete the fields. For more information, see "Report Filters Setup Screen Reference" on page 67.

- 5** Click **Save**.

- 6** Click **Close**.



To test your custom filter fields, see how they appear when you generate your report. Select **Admin > YSR Correspondence > Generate Report** and select your report. Voyager displays your custom filter fields on the left side of the screen.

## Report Filters Setup Screen Reference

The screenshot shows the 'Report Filters Setup' screen. At the top, there is a toolbar with 'Save', 'Preview', and 'Help' buttons. Below the toolbar is a table with columns: Field Name, Label, Sequence, Type, Lookup Name, Multi Select?, Parent, Mandatory?, List Values, Code to ID, Query?, Column View, Show In Title, Inactive?, When Populated In URL, and Delete?. There are five rows in the table, each representing a filter field. The first row is for 'Property' with 'Property' as both the label and field name, sequence 1, and 'Lookup List' as the type. The second row is for 'Tenant' with 'Tenant' as both the label and field name, sequence 2, and 'Lookup List' as the type. The third row is for 'DateFrom' with 'Date From' as both the label and field name, sequence 3, and 'Post Month' as the type. The fourth row is for 'DateTo' with 'Date To' as both the label and field name, sequence 4, and 'Post Month' as the type. The fifth row is empty.

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	List Values	Code to ID	Query?	Column View	Show In Title	Inactive?	When Populated In URL	Delete?
Property	Property	1	Lookup List	ysiPropertyLookup	<input type="checkbox"/>		<input checked="" type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	Allow Edit	<input type="checkbox"/>
Tenant	Tenant	2	Lookup List	ysiTenantLookup	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	Allow Edit	<input type="checkbox"/>
DateFrom	Date From	3	Post Month		<input type="checkbox"/>		<input checked="" type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	Hide	<input type="checkbox"/>
DateTo	Date To	4	Post Month		<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>	Lock	<input type="checkbox"/>
					<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

<b>Field Name</b>	The name of the custom filter field.  The value that you enter here appears when you perform filter mapping.  <b>TIP</b> Report filter field names must differ from the field aliases used in any scripts included in the report. For example, if you use PropID as an alias in any supporting script, do not use PropID as a filter field name.
<b>Label</b>	The label of the custom filter field, visible when generating a report.
<b>Sequence</b>	The sequence in which the custom filter fields appear when you begin the report generation process, and also the number you use to refer to the filter field when mapping filters (if applicable).  For more information on numbered tokens and their use in filtration, see “Named Tokens Versus Numbered Tokens” on page 82 and “Numbered Token Example” on page 84.  <b>NOTE</b> If you implement a parent-child relationship between filters, number the parent filter with a lower number than the child.
<b>Type</b>	The input type of the custom filter field that appears on the report generation screen.  <b>Text</b> Voyager accepts text values in the custom filter field.  <b>List</b> Voyager displays a static or a dynamic list using the values you enter in the <b>List Values</b> field to the report.  <b>Lookup List</b> Voyager displays a lookup list. You can select a standard YSI lookup list from the <b>Lookup Name</b> drop-down list, or you can create a custom lookup list using the <b>List Values</b> field.  <b>TIP</b> For the most part, common YSI lookup lists function correctly in VoyagerPlus. Report designers should however carefully generate and test their reports in VoyagerPlus and, if necessary, replace common YSI lookup lists with custom SQL lookups.  <b>Date</b> Voyager displays a calendar and accepts date values only.  <b>Post Month</b> Voyager displays a calendar and accepts date values only.  <b>Check Box</b> Voyager displays a check box.
<b>Lookup Name</b>	(You can use this field if you select <b>Lookup List</b> in the <b>Type</b> field.)  The pre-defined lookup list that Voyager displays on the report generation screen.  <b>NOTE</b> If you prefer, you can enter a select query in the <b>List Values</b> field instead.
<b>Multi-Select</b>	Enables the user to select multiple values when completing the custom filter fields on the report generation screen.

<b>Parent</b>	Indicates that this filter field is restricted by values in another filter field. Enter the sequence number of the parent filter field. In the preceding graphic, for example, the <b>Tenant</b> field (the child) depends on the <b>Property</b> field.
	<b>TIP</b> A filter field can have multiple parents. For example, you might restrict a lease filter field by properties and tenants. To add multiple parents, enter the sequence number of each parent filter field, separated by a comma (1,2).
	<b>TIP</b> You can also create parent-child relationships between date or number-type filter fields and a lookup list filter field.
	For example, you can create a parent field for batch numbers and child field for a lookup list of control numbers. Voyager shows only the control numbers in the batch the user selects.
	Likewise, ABF Valuations users can create a parent date field and a child field for a lookup list of budgets. Voyager shows only the budgets related to the date the user selects.
<b>Mandatory</b>	Makes the filter a required field on the report generation screen.
<b>List Values</b>	Use this field to enter the values for filter fields of the <b>List</b> or <b>Lookup List</b> type. The field accepts up to 4,000 characters.  If you select <b>List</b> in the <b>Type</b> field, specify the list values separated by a caret sign (^). For example, enter residential^commercial^industrial.  If you select <b>Lookup List</b> in the <b>Type</b> field, you can use this field to enter a custom select statement in lieu of selecting a pre-defined lookup list from the <b>Lookup Name</b> field.  <b>NOTE</b> For more information about how to use custom SQL in this field, see “Defining Dynamic Lists and Custom Lookups” on page 72.
<b>Code to ID</b>	(This field is for use with custom lookup lists only.)  Specify a select query in order to convert the code value to the ID value. The select query must retrieve the respective ID value of each code value.  <b>NOTE</b> For more information about how to use this field with lookup lists, see “Defining Dynamic Lists and Custom Lookups” on page 72.
<b>Query</b>	Select this check box if you are using a dynamic list or a custom lookup list.
<b>Column View</b>	The column in which the custom filter field appears on the report generation screen. You can arrange custom filter fields in up to two columns.
<b>Inactive</b>	Prevents the custom filter from appearing on the report generation screen.

**When Populated in URL** Limits the ability of a user to edit filter values on YSR report generation filters. Applicable only when you access the YSR report via a custom Voyager menu link or custom Voyager button.

**NOTE** This setting has no effect on the YSR report generation filters that you access from the **YSR Generate Report** or **YSR Generate Report New** menus.

**Allow Edit** Users can edit the field on the filter before generating the YSR report. All fields are set to **Allow Edit** by default. If the custom URL specifies a value, the field on the filter populates with that value, but the user can change it.

For example, consider the following URL in which the property is specified:

```
..../pages/CustomCorrespGenerate.aspx?ReportCode=CodeToMatchPI10&Property=comoff01&DateFrom=03/2018&DateTo=06/2018
```

In this example, the **Property** field status is set to **Allow Edit**. In the following graphic, the **Property** field on the filter populates with the value specified in the URL but is still editable.

Pictures Matching Rev R Pix	
Property	comoff01
Tenant	
Date To	06/2018

**Lock** Both the field and value (if indicated in the URL) are visible on the filter. If the custom URL includes a value, the field on the filter populates with that value, but the user cannot change it.

For example, consider the following report URL:

```
..../pages/CustomCorrespGenerate.aspx?ReportCode=CodeToMatchPI10&Property=comoff01&DateFrom=03/2018&DateTo=06/2018
```

In this example, the **Date To** field status is set to **Lock**. In the following graphic, the **Date To** field on the filter populates with the value specified in the URL but is not editable:

Pictures Matching Rev R Pix	
Property	comoff01
Tenant	
Date To	06/2018

**Hide** Hides the field on the filter. If the custom URL specifies a value, the field remains hidden, but Voyager still applies the value when a user generates the report.

In the preceding examples, the custom URL includes a **DateFrom** value, but the **Date From** field status is set to **Hide**. Although the **Date From** field does not show on the filter, the **DateFrom** value is still applied when the report is generated, returning results within a date range from 03/2018 to 06/2018.

**CAUTION** When creating a custom URL, you must specify values for required fields for which the **When Populated in URL** status is set to **Lock** or **Hide**.

## Defining Filter Fields for Use with Voyager Analytics

If your YSR report uses Voyager analytics to retrieve report data, you must take the analytics report filter into consideration when you design the YSR report filter. You may want to reproduce the filters that appear with the analytics in core Voyager, or a sub-set of them. On the other hand, you may want to consider using the filters to:

- Constrain the filter options available to users.

For example, instead of letting your users filter through all the properties in your database, you might create a pick-list that contains only a subset of properties.

- Expand the filter options available to users.

For example, Commercial Analytics for properties includes a filter for Property, Building, and Charge Code. You can use YSR to build an additional filter for Units.

To create an additional filter field for use with a Voyager analytics report in YSR, you must configure YSR to retrieve extra data corresponding to the additional filter field. For example, if the analytics report retrieves property and tenant data but no unit data, you cannot set up a unit filter field unless you modify the analytics. That is, you must configure the analytics report to retrieve an extra column of unit data.



You can retrieve extra data columns only for Voyager analytics that return SQL query strings to YSR, rather than a .Net data table. For more information, see "To add a Voyager analytics report section" on page 43.

You can retrieve extra columns of data by adding a custom token to stand in for the desired data. You can add custom tokens to the analytics report on the **Report Sections Setup** screen. For more information, see "To add a Voyager analytics report section" on page 43. See also "Report Sections Setup Screen Reference" on page 103.

## Defining Filter Fields for Data Range

This section describes how to set up custom filter fields for a range of data (such as a date range). There are two steps to creating range filters: filter-definition and filter-mapping.

This section provides two examples of setup screens for custom SQL-driven reports.

- Basic filter range, mapped to custom SQL script

Commercial Charges	
<b>Basic filter range</b>	<input type="text" value="Property"/> <input type="text" value="Tenant"/> <input type="text" value="Date From"/> <input type="text" value="01/01/2014"/> <input type="button" value="..."/> <input type="text" value="To"/> <input type="text" value="02/28/2014"/> <input type="button" value="..."/>
	Report Name: <input type="text" value="Commercial Charges (CommChg)"/> <input type="button" value="..."/> Output Type: <input type="button" value="Screen"/> <input type="button" value="Grid"/> Merge Reports: <input type="checkbox"/> Show Grid: <input type="checkbox"/> <input type="button" value="Generate"/> <input type="button" value="Clear"/>
	Attach Reports: <input type="checkbox"/> Email Reports: <input type="checkbox"/> Publish To SharePoint: <input type="checkbox"/> Show on Portal: <input type="checkbox"/>

- Linked filter range, mapped to custom SQL script



If your report section uses a Voyager analytics report to retrieve data, set up your YSR report using a basic filter range rather than a linked filter range. The filter mapping process for Voyager analytics filters requires separate (basic) filter fields, not linked fields.

### Basic Filter Range Setup Example

**Filter-definition** To create a basic filter range, add **From** and **To** filter fields following normal setup steps.

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?
Property	Property	1	Lookup List	ysiPropertyListLookup	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Tenant	Tenant	2	Lookup List	ysiTenantLookup	<input type="checkbox"/>	1	<input type="checkbox"/>
DateFrom	Date From	3	Date		<input type="checkbox"/>		<input type="checkbox"/>
DateTo	Date To	4	Date		<input type="checkbox"/>		<input type="checkbox"/>
					<input type="checkbox"/>		<input type="checkbox"/>

**Filter-mapping** On the **Map Script Filter** or **Map Standard Analytics Filter** screen, use the logical operators **>=** and **<=** to complete the **Report Filter Condition** field.

YSR Report Filters	Script Filter Condition
Property (#Condition1#)	p.hmy in (#Property#)
Tenant (#Condition2#)	t.hmyperson in (#Tenant#)
DateFrom (#Condition3#)	ckd.dtdate >= ('#DateFrom#')
DateTo (#Condition4#)	ckd.dtdate <= ('#DateTo#')



Depending on the type of data the filter points to, you may need to use a convert expression. For example, you may need to convert text data to numeric data.

## Linked Filter Range Setup Example

**Filter-definition** To create a linked filter range, use a colon separator to create two filter fields on one line.

The screenshot shows the 'Report Filters Setup' dialog box. At the top, it says 'YSR Report: Comm Charges Demo (CommCh)'. Below that are 'Save', 'Preview', and 'Help' buttons. The main area contains a table with columns: Field Name, Label, Sequence, Type, Lookup Name, Multi Select?, Parent, and Mandatory?. There are three rows: 'Property' (Label: Property, Sequence: 1, Type: Lookup List, Lookup Name: ysiPropertyListLookup, Multi Select?: checked, Mandatory?: checked), 'Tenant' (Label: Tenant, Sequence: 2, Type: Lookup List, Lookup Name: ysiTenantLookup, Multi Select?: unchecked, Parent: 1, Mandatory?: unchecked), and 'DateFrom:DateTo' (Label: Date From:To, Sequence: 3, Type: Date, Lookup Name: (empty), Multi Select?: unchecked, Parent: (empty), Mandatory?: unchecked).

**Filter-mapping** On the **Map Script Filter** or **Map Standard Analytics Filter** screen, use the logical operator *between* to complete the **Report Filter Condition** field.

For example: ckd.ddate between '#DateFrom#' and '#DateTo#'

The screenshot shows the 'Map Script Filter' dialog box. It has 'YSR Report: Comm Charges Demo (CommCh)' and 'Report Code: commch' at the top. Below are 'Save', 'Close', and 'Help' buttons. The main area has two sections: 'YSR Report Filters' and 'Script Filter Condition'. The filters listed are: 'Property (#Condition1#)', 'Tenant (#Condition2#)', and 'DateFrom:DateTo (#Condition3#)'. The corresponding script conditions are: 'p.hmy in (#Property#)', 't.hmyperson in (#Tenant#)', and 'ckd.ddate between '#DateFrom#' and '#DateTo#'. The 'DateFrom:DateTo' row is partially cut off.

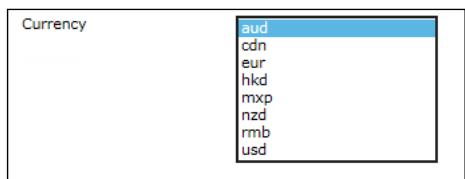


Depending on the type of data the filter points to, you may need to use a convert expression. For example, you may need to convert text data to numeric data.

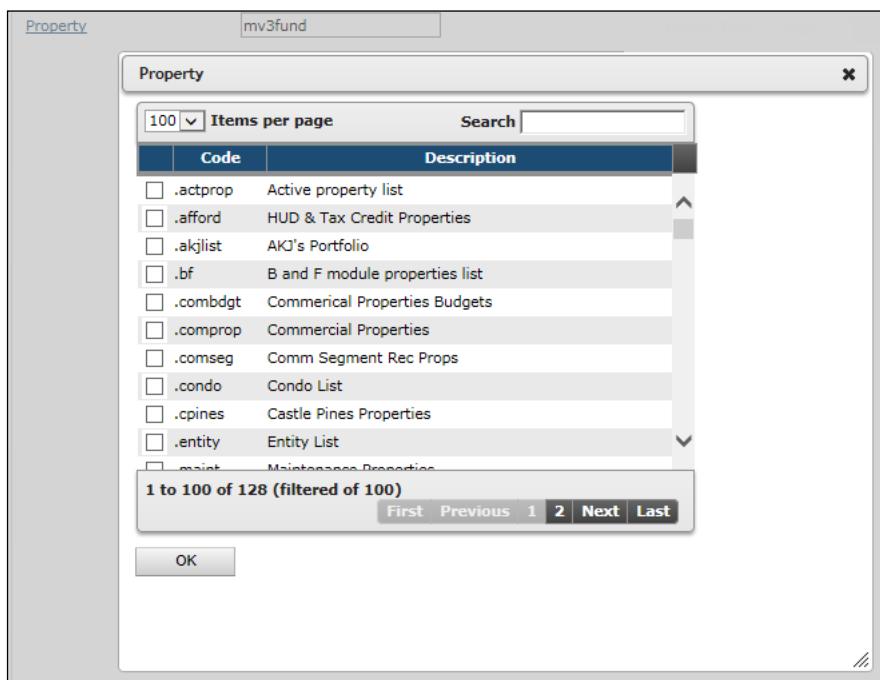
## Defining Dynamic Lists and Custom Lookups

When you define the custom filter fields that appear on the report generation screen, you can create dynamic drop-down lists and custom lookup lists by using short SQL queries.

**Dynamic drop-down lists** present to the user a list of options available in your database at run time. For example, the following graphic shows a dynamic drop-down list where a user can select a currency at run time.



**Custom lookup lists** provide the user with a link to a search screen where they can search for a specific object. The following graphic shows an example custom lookup list.



Each type of list has specific syntax requirements:

- Dynamic lists must return two columns of data aliased **TextField** and **ValueField**.
- Custom lookup lists must return two columns of data aliased **CodeField** and **TextField**.

#### Sort order of lists

You can control the sort order of your lists by adding an **order by** clause to your SQL query. For example, use the following string to order a list by descending codes: **order by CodeField desc**.

## This section includes the following procedures:

To define a dynamic drop-down list .....	74
To define a custom lookup list .....	74

### To define a dynamic drop-down list

- 1 Select **Admin > YSR Correspondence > Setup Report**. Complete the filter and click **Submit**.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see “Adding the YSR Admin Menu” on page 162.

- 2 Click **Define Filters**. The **Report Filters Setup** screen appears.

- 3 In the row corresponding to the dynamic list, select **List** in the **Type** field.
- 4 In the same row, select the **Query** check box.
- 5 In the same row, specify a select statement in the **List Values** field.



The query must return two columns of data aliased **TextField** and **ValueField**. For example:

```
Select SCODE TextField, HMY ValueField From INTCURRENCY_INFO
```

- 6 Click **Save**.

### To define a custom lookup list

- 1 Select **Admin > YSR Correspondence > Setup Report**. Complete the filter and click **Submit**.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see “Adding the YSR Admin Menu” on page 162.

- 2 Click **Define Filters**. The **Report Filters Setup** screen appears.
- 3 In the row corresponding to the custom lookup list, select **Lookup List** in the **Type** field.
- 4 In the same row, select the **Query** check box.
- 5 In the same row, specify a select statement in the **List Values** field.



The query must return two columns of data, aliased **CodeField** and **TextField**. The column aliased **CodeField** must contain the unique identifier of the data set (an scode, for example, is suitable in many, but not all, tables).



For example, the following select statement obeys the constraints for a lookup list without a parent lookup:

```
SELECT p.sCode CodeField, p.sAddr1 TextField FROM Property p WHERE 1=1
```



The following select statement obeys the constraints for a lookup list with a parent lookup, where the parent is the filter field with sequence number 01:

```
SELECT p.sCode CodeField, p.sAddr1 TextField, o.uCode Parent01Field
FROM Property p
Inner Join Owner o On o.hMyPerson=p.hLegalEntity
WHERE 1=1
```

#### 6 In the same row, specify a select statement in the **Code to ID** field.



The query must return unique identifiers (the hmy column, for example) aliased **IDField**. The query must also contain a substitutable reference to the **CodeField** from the previous select statement in the WHERE clause. For example:

```
SELECT hMy IDFfield FROM Property WHERE 1=1 AND sCode in ([CodeField])
```

In this example, CodeField is the alias for sCodes in the previous select.

#### 7 Click **Save**.

## Displaying User's Filter Criteria in Title of Report (Dynamic Titles)

In some cases you may want the title of a YSR report to display the filter criteria that a user enters at runtime. For example, suppose a user generates a report using the filter criteria depicted here:

<u>Property</u>	comoff01	Report Name	Comm Charges (CommCh)
<u>Tenant</u>	t0000016	Output Type	Excel
Date	12/31/2008	Merge Reports	<input type="checkbox"/>
		Show Grid	<input type="checkbox"/>
		Attach Reports	<input type="checkbox"/>
		Email Reports	<input type="checkbox"/>
		Show on Portal	<input type="checkbox"/>
		<b>Generate</b>	<b>Clear</b>

You can design your report and report template so that the user's filter terms appear in the report title.

<b>Comm Charges</b> <u>Property</u> comoff01 <u>Tenant</u> t0000016 <u>Date</u> 12/31/2008			Report Name	Comm Charges (CommCh)															
			Output Type	Screen															
			Merge Reports	<input type="checkbox"/>															
			Show Grid	<input type="checkbox"/>															
			Attach Reports	<input type="checkbox"/>															
			Email Reports	<input type="checkbox"/>															
			Show on Portal	<input type="checkbox"/>															
			<b>Generate</b>	<b>Clear</b>															
<b>Report title</b> <u>Monthly Activity Report for</u> property: comoff01 . tenant: t0000016																			
<table border="1"> <thead> <tr> <th>Property</th> <th>Tenant</th> <th>Date</th> <th>Monthly Rent</th> <th>CAM</th> </tr> </thead> <tbody> <tr> <td>comoff01</td> <td>Fidelity Escrow - Santa Monica</td> <td>12/31/2008</td> <td>15958.8</td> <td>2291.52</td> </tr> <tr> <td></td> <td>Fidelity Escrow - Santa Monica</td> <td>12/31/2008</td> <td>15958.8</td> <td></td> </tr> </tbody> </table>					Property	Tenant	Date	Monthly Rent	CAM	comoff01	Fidelity Escrow - Santa Monica	12/31/2008	15958.8	2291.52		Fidelity Escrow - Santa Monica	12/31/2008	15958.8	
Property	Tenant	Date	Monthly Rent	CAM															
comoff01	Fidelity Escrow - Santa Monica	12/31/2008	15958.8	2291.52															
	Fidelity Escrow - Santa Monica	12/31/2008	15958.8																

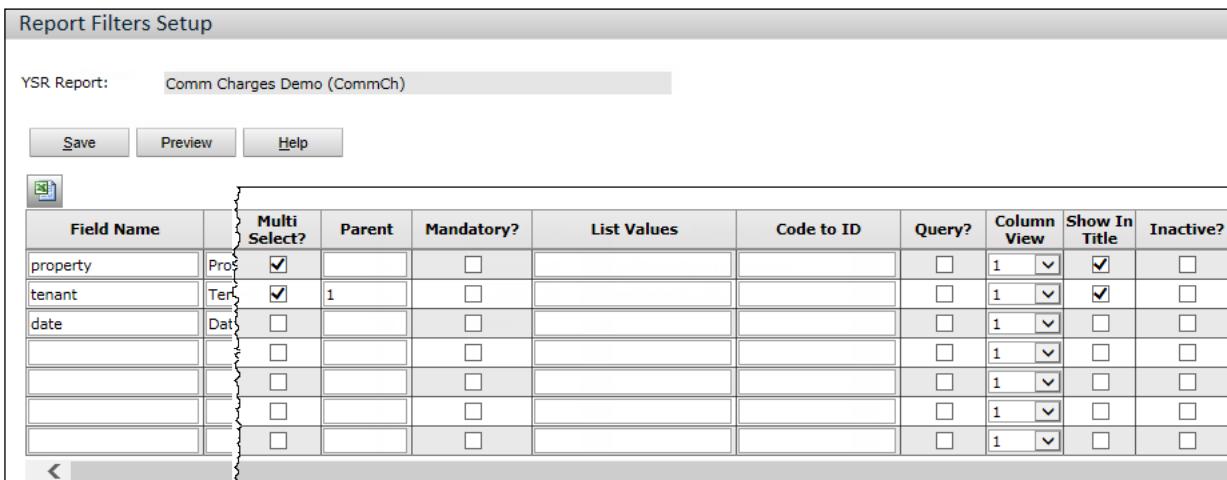
To display report titles, YSR has a built-in report section called **YSRFilterValues** and a flag (**Show in Title**) on the **Report Filters Setup** screen (where you define your YSR report filter).

You do not need to complete any section mapping or filter mapping to use the YSRFilterValues report section. You can simply add YSRFilterValues smart markers to your report templates. Supported smart markers include:

<code>&amp;=YSRFilterValues.ReportTitleValues</code>	Displays the user's filter values for all the filter fields with <b>Show in Title</b> selected on the <b>Report Filters Setup</b> screen.
<code>&amp;=YSRFilterValues.FilterFieldName</code>	<p>Displays the user's filter value for one specific filter field, referenced by its field name on the <b>Report Filters Setup</b> screen.</p> <p>For example, suppose your YSR report filter has a filter field with the field name <b>property</b>. You can add this smart marker to your template:</p> <pre>&amp;=YSRFilterValues.property</pre> <p>At run time, Voyager replaces the smart marker with the property entered by the user.</p>

### To display a user's filter criteria in the title of a report

- 1 Navigate to the **Report Filters Setup** screen.
- 2 Select the **Show in Title** check box corresponding to the filter values you want to display in the report title.



- 3 Click **Save**.
- 4 Edit your report template:
  - If you are using Excel, add the following smart marker: `&=YSRFilterValues.ReportTitleValues`

A	B	C
1		
2	<b>Monthly Activity Report for</b>	<code>&amp;=YSRFilterValues.ReportTitleValues</code>
3		

- b** If you are using Word, add the following merge field: <<ReportTitleValues>>

- 5** Save your report template to the Report path and generate your report.



Voyager displays the filter field name followed by the user's filter term.

If the user does not complete one or more of the filter fields, the filter field name does not appear in the report title.

## Filter Mapping for Voyager Analytics Report Sections

This section provides an overview of the filter mapping process for YSR reports that use Voyager analytics to retrieve data.

If you select a Voyager analytics report as the data source for a YSR report, then the filter fields associated with the analytics engine become exposed to YSR. For example, the YSR Trial Balance report (an example YSR report included in the Financial Analytics Plug-in) uses Voyager Financial Analytics to retrieve data. The following graphic shows all the filter fields native to Financial Analytics.

If you are building a YSR report that uses Financial Analytics, however, you probably do not need to expose all the Financial Analytics filter fields to the user. The YSR Trial Balance report, for example, displays only the filter fields necessary for generating the Trial Balance report. The YSR Trial Balance report offers only the custom filter fields shown in the following graphic:

Each of the custom filter fields above is part of the YSR report design. The **Property** field leverages a standard lookup list, `ysiPropertyListOrLookup`; the **Book** field uses another standard lookup list, and so on, as illustrated in the next graphic.

Report Filters Setup									
YSR Report: Trial Balance (TrialBal)									
<input type="button" value="Save"/> <input type="button" value="Preview"/> <input type="button" value="Help"/>									
									
Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	List Va	
PropertyCode	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
Book	Book	2	Lookup List	ysiBookLookup	<input type="checkbox"/>		<input type="checkbox"/>		
TreeCode	Account Tree	3	Lookup List	ysiAccountTreeLookup	<input type="checkbox"/>		<input checked="" type="checkbox"/>		
FromPeriod	From Period	4	Post Month		<input type="checkbox"/>		<input type="checkbox"/>		
ToPeriod	To Period	5	Post Month		<input type="checkbox"/>		<input type="checkbox"/>		
IsConsolidate	Consolidate	6	Checkbox		<input type="checkbox"/>		<input type="checkbox"/>		

The filter mechanisms proper to the custom filter fields are entirely distinct from the filter mechanisms proper to Financial Analytics. In order to coordinate filtration, therefore, you must complete the filter mapping process.

When you map filters for a YSR report that uses Voyager analytics, you can:

- Map each custom filter field to a field in the Voyager analytics report filter.
- Hard-code elements of the Voyager analytics report filter that are essential to the report design.
- Ignore elements of the Voyager analytics report filter that are unrelated to your report.

To illustrate, the following graphic shows the **Map Standard Analytics Filter** screen for the YSR Trial Balance report.

Map Standard Analytics Filter		
YSR Report: Trial Balance (TrialBal)		
<input type="button" value="Save"/> <input type="button" value="Close"/> <input type="button" value="Help"/>		
Standard Analytics Report Filters	YSR Report Filters	Constant Value
FinType		TRIAL BALANCE
SuppressZero		1
TreeLevel		2
PropertyCode	PropertyCode	
BookCodeList	Book	
TreeCode	TreeCode	
FromMMYY	FromPeriod	
ToMMYY	ToPeriod	
IsConsolidate	IsConsolidate	

- The Financial Analytics is hard-coded to generate the Trial Balance report type only (row 1).

- The **PropertyCode**, **BookCodeList**, **TreeCode**, **FromMMYY**, and **ToMMYY** fields of the Financial Analytics report filter are mapped to custom YSR filter fields (rows 3 - 7).
- The majority of the Financial Analytics report filter elements are ignored because they are unnecessary or unrelated to the Trial Balance report.

To set up a YSR report that uses Voyager analytics report data, therefore, you must take the filter mapping process into consideration from the beginning of the report design process. You must add the custom filter fields that are appropriate to your report, and you must map them to the corresponding fields in the Voyager analytics report filter. You must also be familiar with the acceptable values for the Voyager analytics report filter. You must know, for example, whether a check box accepts **0** and **1** or **True** and **False**.



For a list of possible values for the **Constant Value** column, see Appendix D, “Constant Values for Analytics Data Source Filter Mapping.”

For more information about the requirements of a specific Voyager analytics, contact Yardi technical support.

### To map filters for a Voyager analytics YSR report

- Select **Admin > YSR Correspondence > Setup Report**. The **YSR Correspondence Setup** screen appears.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see “Adding the YSR Admin Menu” on page 162.

- Click **Submit**. The **YSR Report Setup** screen appears.

The screenshot shows the YSR Report Setup screen with the following details:

- Code:** TrialBal
- Description:** Trial Balance
- Top-Level Select Statement:** select p.hmy propertyId,p.scode sCode from property p where
- Key Column:** propertyId
- Inactive:**
- Notes:**
  - FinType : TRIAL BALANCE
  - SuppressZero : 1
  - TreeLevel : 2
  - Filters : PropertyCode,TreeCode,FromPeriod,ToPeriod, Book, TreeCode, IsConsolidate
- Buttons:** Save, New, Save Copy, Define Filters, Map Top-Level Filter, Dump SQL, Delete Setup, Help
- Tab Bar:** Report Setup (selected), Attachment & Email, Additional Attachment & Email, Output Options, Digital Signature
- Report Setup Table:**

Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?
10	TrialBal	YSR_TrialBalance.xlsx		propertyId	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- 3 On the **Report Setup** tab, in the row corresponding to the sub-report that uses Voyager analytics, click **Sections**. The **Report Sections Setup** screen appears.

The screenshot shows the 'Report Sections Setup' dialog box. At the top, it displays 'YSR Report: Trial Balance (TrialBal)' and 'Report Code: TrialBal'. Below this are 'Save', 'Close', and 'Help' buttons. The main area contains two tabs: 'Report Sections' (selected) and 'Attachment Setup'. Under 'Report Sections', there is a table with columns: Section Code, Description, SELECT Name, Standard Analytics Report, Map Standard Analytics Filter, Tokens, Relations, Key Columns, Primary?, and Multiple Rows?. One row is visible in the table:

Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	Primary?	Multiple Rows?
TrialBal	Trial Balance		Financial Analytics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	propertyId	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>

- 4 In the row or rows corresponding to each report section that uses Voyager analytics, click **Map Standard Analytics Filter**. The **Map Standard Analytics Filter** screen appears.

The screenshot shows the 'Map Standard Analytics Filter' dialog box. At the top, it displays 'YSR Report: Trial Balance (TrialBal)', 'Report Code: TrialBal', and 'SectionCode: TrialBal'. Below this are 'Save', 'Close', and 'Help' buttons. The main area contains a table with columns: Standard Analytics Report Filters, YSR Report Filters, and Constant Value. Several rows are listed in the table:

Standard Analytics Report Filters	YSR Report Filters	Constant Value
FinType		TRIAL BALANCE
SuppressZero		1
TreeLevel		2
PropertyCode	PropertyCode	
BookCodeList	Book	
TreeCode	TreeCode	
FromMMYY	FromPeriod	
ToMMYY	ToPeriod	
IsConsolidate	IsConsolidate	

- 5 Complete the fields.
- In the **Standard Analytics Report Filters** column, select the elements of the Voyager analytics report filter that are important for your report.
  - Use the **Constant Value** column to hard-code any elements of the Voyager analytics report filter that you want to remain constant.
  - Use the **YSR Report Filters** column to map Voyager analytics filter fields to the custom filter fields that the YSR user completes at run time. (The **Report Filter** column displays the field names of your YSR report).
- 6 Click **Save**.

# Filter Mapping for Scripted Report Sections

## In this section:

Named Tokens Versus Numbered Tokens .....	82
Named Token Example.....	84
Numbered Token Example .....	84
Filtering the Top-Level Select Statement.....	86
Writing WHERE Clauses .....	87

This section provides an overview of the filter mapping process for YSR reports that use custom SQL scripts to retrieve data.

When you use a custom SQL script to retrieve data for your report, you must provide a way for Voyager to filter data. In a SQL script, you can filter data by adding a WHERE clause, like WHERE p.scode = 'comoff01'. In Yardi terminology, the string that follows the WHERE keyword is also known as the Value 1 parameter. It is rare to hard-code a Value 1 parameter directly into a YSR report. In almost all cases, it is preferable to use tokens in the Value 1 parameter to represent the values the user supplies at run-time.

## Tokens

You can use named tokens like (#Property#) or numbered conditions like #condition1# in your script's WHERE clause to represent the user's filter values. If you use named tokens, you can pass the user's filter values directly to an underlying script without completing **Map Filter** screens. If you use numbered tokens, you must also map your filters inside the YSR report design, using the appropriate **Map Filter** screen or screens. Numbered tokens are advantageous in that they support more modular report design. You can more easily reuse a script with numbered tokens in another YSR report.

For more information, see "Named Tokens Versus Numbered Tokens" on page 82.



YSR scripts can also contain user and session-related tokens. For more information, see "Session and User-Related Tokens" on page 52.

## The Value 1 Parameter

Reflecting its genesis in the formal conventions of a Yardi SQL script, the string that follows the WHERE keyword is called the Value 1 (Val 1) parameter. In a traditional Yardi script, you can specify a Val 1 parameter (like **p.hmy in #Property#**) in the //Filter section of the script. Suppose you specify the Val 1 parameter on sequence line 1 of the //Filter section. Then you can add the #Condition1# operator (corresponding to sequence 1) to the WHERE clause of a //Select section of the script. When the report renders, Voyager replaces the #Condition1# operator with the Val 1 parameter by appending (ANDing) it into the WHERE clause.

To accommodate the act of ANDing, WHERE clauses in Yardi scripts typically appear as follows:

WHERE 1=1 #Condition1#

YSR scripts do not recognize text included in any //Filter section, but YSR uses a similar system of substitution to pass the user's filter criteria to the underlying SQL select sections. Depending on whether you use named or numbered tokens in the underlying SQL script, you must add the Val 1 parameter directly to your SQL select statement, or you can enter it on the corresponding **Map Script Filter** screen.

For more information about using the Val 1 parameter in YSR reports, see the following topics.

For more information about the Val1 parameter and its role in linking the //Filter and //Select sections of conventional Yardi scripts, see Chapter 5, "Vista Filters" in the *Yardi SQL Scripting Guide*.



Value 2 parameters are not supported in YSR.

A Value 2 parameter is a construct used in the //FILTER section of a conventional Yardi SQL Script. The Val 2 parameter is a portion of SQL that the scripting engine applies, as a filter condition, in cases when a user leaves a non-mandatory filter field blank.

## Named Tokens Versus Numbered Tokens

There are two types of tokens you can use in a SQL script to stand for user-supplied values: named tokens and numbered tokens.

**Named tokens** use the field name of the custom filter field whose value they represent. For example, suppose you have a YSR report with the following custom filter fields:

Report Filters Setup								
YSR Report: Comm Charges Demo (CommCh)								
<input type="button" value="Save"/> <input type="button" value="Preview"/> <input type="button" value="Help"/>								
Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	
property	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
tenant	Tenant	2	Lookup List	ysiTenantLookup	<input checked="" type="checkbox"/>	1	<input type="checkbox"/>	
date	Date	3	Date		<input type="checkbox"/>		<input type="checkbox"/>	

Named tokens for this report include (#property#), (#tenant#), and (#date#).

Named tokens resolve to the value returned by the filter field, with no additional syntax. For example, suppose the user enters comoff01 into the **Property** field defined in the preceding graphic. Suppose also that the hmy of comoff01 is 2. The following table shows how Voyager resolves the (#property#) token when rendering the report:

Original SQL syntax	Resolved syntax
WHERE p.hmy in (#property#) (correct syntax)	WHERE p.hmy in (2)

Original SQL syntax	Resolved syntax
WHERE (#property#) (incorrect syntax)	WHERE (2) (SQL error)
WHERE 1=1 and (#property#) (incorrect syntax)	WHERE 1=1 and (2) (SQL error)
WHERE 1=1 (#property#) (incorrect syntax)	WHERE 1=1 (2) (SQL error)

**Numbered tokens** refer to the sequence number of the custom filter fields whose value they represent. Numbered tokens for the report filter illustrated in the previous section include #condition1#, #condition2#, and #condition3#.



You can also use #conditions# to stand for all the filter fields in the report. #Conditions# appends all Val1 parameters of all defined filter elements. Non-mandatory filter elements that are not submitted by the user are ignored.

When you use numbered tokens in your SQL script, you must supply additional information in your YSR report design. You must use the **Map Filter** screen to supply the portion of the WHERE clause that Voyager appends (ANDs) to your script at runtime. In Yardi terminology, this clause is known as the Value 1 parameter.

The following table shows possible combinations of numbered conditions and Value 1 parameters and their results.

Original SQL syntax	Val 1 Parameter on Map Filter	Resolved SQL syntax
WHERE 1=1 #condition1# (correct syntax)	p.hmy in (#property#) (correct syntax)	WHERE 1=1 and p.hmy in (2)
WHERE 1=1 #condition1# #condition2# (correct syntax)	p.hmy in (#property#) t.hmyperson in (#tenant#) (correct syntax)	WHERE 1=1 AND p.hmy in (2) and t.hmyperson in (96)
WHERE 1=1 #conditions# (acceptable syntax)	p.hmy in (#property#) t.hmyperson in (#tenant#) (correct syntax)	WHERE 1=1 AND p.hmy in (2) and t.hmyperson in (96)
WHERE 1=1 #condition1# (correct syntax)	no mapping (blank <b>Map Filter</b> screen)	WHERE 1=1 #condition1# (SQL error)
WHERE 1=1 #condition1# (correct syntax)	p.hmy in (#condition1#) (incorrect syntax)	WHERE 1=1 (no error, but Voyager does not filter data.)

WHERE 1=1 (#condition1#)	p.hmy in (#property#)	WHERE 1=1 (AND p.hmy in (2))
(incorrect syntax)	(correct syntax)	(SQL error)

## Named Token Example

If you are accustomed to writing Yardi scripts, the most straightforward way to manage filtration is to used named tokens (like (#Property#) or (#Tenant#)) in your Val 1 parameter and place the Val 1 parameter directly in your script, as illustrated in the following graphic:

The Val 1 parameter refers to the YSR report's property filter by its field name, **Property**.

If the YSR report has additional custom filter fields in its report filter, you can add the additional Val 1 parameters to the script using **and**. For example: and t.hmyperson in (#Tenant#) and ckd.dtdate in

The disadvantage to placing the Val 1 parameter directly in your script is that your report is less flexible than it could be. You cannot easily re-use your script in another YSR report, as you must adapt it to the custom filter fields and naming conventions of other reports.

## Numbered Token Example

The most modular approach to building YSR reports is to remove the Val 1 parameter from all underlying scripts and include it, instead, inside of the YSR report design. This is the most flexible method of report design, and it makes it easy to swap out one SQL script for another or re-use the same SQL script in multiple reports.

To make the Val 1 parameter part of the YSR report design, you must use numbered conditions like #condition1# and #condition2# in your script. You can also use #conditions# to stand for all the filter fields in the YSR report filter. Then you must add the Val 1 parameter or parameters to the corresponding **Map Filter** screen.



#Conditions# appends all Val1 parameters of all defined filter elements. Non-mandatory filter elements that are not submitted by the user are ignored.

For example, suppose you have a report for retrieving commercial lease data. The YSR report filter contains the following custom filter fields (note the field name and sequence number of each field):

**Report Filters Setup**

YSR Report: Comm Charges Demo (CommCh)

**Save** **Preview** **Help**

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent
property	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>	
tenant	Tenant	2	Lookup List	ysiTenantLookup	<input checked="" type="checkbox"/>	1
date	Date	3	Date		<input type="checkbox"/>	
					<input type="checkbox"/>	

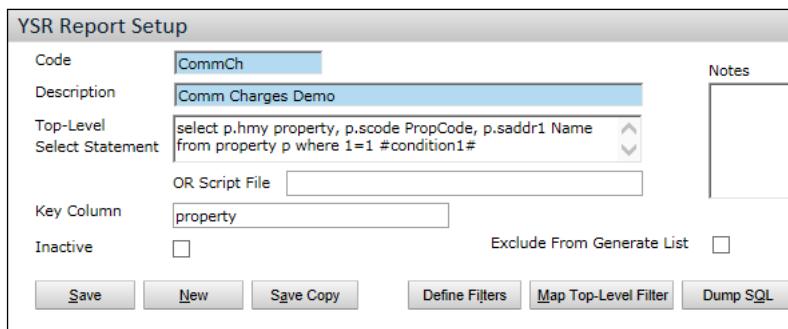
The SQL script in the underlying sub-report uses numbered tokens, and the Val 1 parameters are included on the corresponding **Map Filter** screen.

When the user generates the report, Voyager appends the Val 1 parameters that appear on the **Map Filter** screen to the SQL script, using **AND**.

In this illustration, Voyager resolves the WHERE clause in the script as follows:

```
WHERE 1 = 1 and p.hmy in (#property#) and t.hmyperson in (#tenant#) and ckd.dtdate
```

YSR uses the same filtration conventions for top-level select statements as it does for underlying sub-reports. You can use named tokens and add the Val 1 parameter directly to the top-level select, as in the following graphic:



Alternatively, you can use numbered conditions in the top-level select statement and add Val 1 parameters to the corresponding **Map Filter** screen.

The screenshot shows the 'YSR Report Setup' dialog box and a separate 'Map Top-Level Filter' dialog box. In the Report Setup, the 'Top-Level Select Statement' is: 'select p.hmy property, p.scode PropCode, p.saddr1 Name from property p where 1=1 #condition1#'. In the Map Top-Level Filter dialog, there is one filter entry: 'property (#Condition1#)' is mapped to 'p.hmy in (#property#)'.

Both methods are acceptable ways of filtering data for the top-level select statement.

## Writing WHERE Clauses

To write effective WHERE clauses for your YSR report design, you must be familiar with the custom filter fields in the YSR report filter. You must know the field names of each filter field, their sequence number, and a little bit about how they work.

For example, the standard `ysiPropertyOrListLookup` works differently than a simple pick-list of properties like `comoff01^comoff02^comind01^comind02`. The `ysiPropertyOrListLookup` retrieves properties by their property `hmy`, while the simple pick-list displays the list values themselves. The appropriate `Val 1` parameter for the `WHERE` clause for the former is `p.hmy in (#FilterFieldName#)`. The appropriate `Val 1` parameter for the latter might be something like `p.scode in (#FilterFieldName#)`, depending on the conventions used in the underlying SQL.

If you use any of the standard YSI lookups, the `Val 1` parameter obeys the conventions of the lookup list. For more information about the `Val 1` parameters for commonly used lookups, see “`Val1 Parameter Conventions`” on page 272.

If you use any other method when designing the YSR report filter fields, then the `Val 1` parameter must respect the conventions of the underlying SQL script or scripts. For example, suppose you use the following script to retrieve commercial lease data from the `CommKPIDetails` table:

```
//SELECT Charges
Select distinct
    p.hmy                      prophmy
    ,p.scode                     property
    ,t.slastname                 tenant
    ,ckd.hunit                   unit
    ,ckd.dtdate                  date
    ,ckd.dLeaseMonthlyRent       MonthlyRent
    ,ckd.dLeaseMonthlyCam        MonthlyCAM
    ,ckd.hmy                      ckdhmy

from CommKPIDetail ckd
inner join property p on ckd.hprop = p.hmy
inner join tenant t on ckd.htenant = t.hmyperson

where 1=1 #conditions#
//END SELECT
```

Suppose also that your report filter has a filter field with field name **Date**. To link the Date filter field to the dates retrieved by the SQL illustrated above, use the following `Val 1` parameter:

`ckd.date`

Notice that the `Val 1` parameter observes the naming conventions used in the underlying script (`ckd.date`) and the naming convention of the custom filter field (Date).

## CHAPTER 4

# YSR Report Section and Template Setup

### In this chapter:

Report Section and Template Setup Overview .....	89
Report Section Setup .....	101
Report Template Setup.....	108
Dynamic Template Selection .....	114

This chapter describes how to set up report sections and report templates.

## Report Section and Template Setup Overview

### In this section:

Report Section Example .....	90
Excel Template Example.....	91
Word Template Example .....	93
Nested Tables in Word .....	94
Working with Word 2013.....	99
PDF Template Example.....	100
Report Output Example.....	101

*Report section* setup and *report template* setup go hand-in-hand.

Report sections are the individual data sources that power a report. A report section can be a single named SELECT section in a custom SQL script or a single Voyager analytics data source. Every YSR sub-report has at least one report section.

Report templates are the Excel, Word, and PDF files that become the final report documents in a YSR report. They contain text and formatting, and they use placeholders to stand in for Voyager data. Excel files use smart markers as placeholders, Word files use merge fields, and PDF files use tokens. Each placeholder stands in for data from one section of the report.

When you generate a YSR report, Voyager matches the data returned by each report section with the placeholders in your report templates. The placeholders refer to Voyager data by section and by the field names (or aliases) used by the script or Voyager analytics data source that powers the section.



Knowing the field names used in your report sections is critical for creating the smart markers (Excel), merge fields (Word), or tokens (PDF) for use in your report templates.

If you are working with a custom SQL script, you can simply review your script to identify field names. If you are working with Voyager analytics data sources, use the Yardi Excel Add-In to identify field names. For more information about the Add-In, see Chapter 8, "Yardi Excel Add-In for YSR."

In some cases you can also force YSR to generate a SQL error that reveals field names. For more information about this method, see "Identifying Field Names for Voyager Analytics Reports" on page 108.

## Report Section Example

Report sections are the most granular unit of YSR report design, corresponding to one data source. Every sub-report in a YSR report has at least one report section, and may have multiple report sections.

There are several reasons that a YSR report might contain multiple sections. For example, the report:

- Contains data from multiple sources.
- Contains both header information (single data items) and detail information (expanding row data).
- Re-uses some but not all sections of similar YSR report (uses modular report design).

For example, the following graphic shows the **Report Section Setup** screen for a YSR sub-report that contains three report sections. Most important (and most typical) are the header and detail sections, coded **header** and **charge**, respectively.

Report Sections Setup																																																									
YSR Report:	Credit Note Domestic (creditND)																																																								
Report Code:	cnd																																																								
<input type="button" value="Save"/>	<input type="button" value="Close"/>	<input type="button" value="Help"/>																																																							
<input type="button" value="Report Sections"/> <input type="button" value="Attachment Setup"/>																																																									
<table border="1"> <thead> <tr> <th colspan="2"></th> <th>Section Code</th> <th>Description</th> <th>SELECT Name</th> <th>Standard Analytics Report</th> <th>Map Standard Analytics Filter</th> <th>Tokens</th> <th>Relations</th> <th>Key Columns</th> </tr> </thead> <tbody> <tr> <td colspan="2"></td> <td>header</td> <td>header</td> <td>header</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>TenantHmy</td> </tr> <tr> <td colspan="2"></td> <td>charge</td> <td>charges</td> <td>DomChargeQuery</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>TenantHmy</td> </tr> <tr> <td colspan="2"></td> <td>total</td> <td>total</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>TenantHmy</td> </tr> <tr> <td colspan="2"></td> <td></td> <td></td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> </tr> </tbody> </table>										Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns			header	header	header	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy			charge	charges	DomChargeQuery	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy			total	total		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy						<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns																																																
		header	header	header	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy																																																
		charge	charges	DomChargeQuery	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy																																																
		total	total		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy																																																
					<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																																	

Header section

Details section

In this sub-report, multi-row detail data in the **charge** section comes from a SELECT section named **DomChargeQuery**. Data in the **header** section comes from a SELECT section named **header**. The sub-report does not contain Voyager analytics data.



If a report section uses a Voyager analytics data source rather than a custom SELECT statement, the **SELECT Name** field is blank and the name of the data source appears in the **Standard Analytics Report** field.

For a full definition of all the fields on the **Report Section Setup** screen, see “Setting up Report Sections” on page 101.

To review an example of an Excel template that uses these report sections, see “Excel Template Example” on page 91.

To review an example of a Word template that uses these report sections, see “Word Template Example” on page 93.

To review example report output based on these report sections, see “Report Output Example” on page 101.

## Excel Template Example

This section provides an example of an Excel report template that displays data from the report sections illustrated in the preceding topic.

This template uses data from the **charge** and **header** sections, both of which correspond to named SELECT sections in a custom SQL script. In place of Voyager data, the template uses smart markers as stand-ins. Smart markers use this syntax: &=sectioncode.fieldname.

CREDIT NOTE				
TENANT	&=Header.TenantName &=Header.BillingFirstName  &=Header.BillingLastName	GST REG NO.	M9-0010744 A  INVOICE DATE  LEASE START  LEASE END	&=Header.InvoiceDate  &=Header.TenantFromDate  &=Header.TenantToDate
PROPERTY NAME	&=Header.PropertyName	UNIT NO	&=Header.Units	
TENANT CODE	&=Header.TenantCode			
DUE DATE	DESCRIPTION	FROM	TO	NET AMOUNT (SGD)
&=Charge.DUEDATE	&=Charge.ChargeDesc	&=Charge.ChargeDateFrom	&=Charge.ChargeDateTo	&=Charge.NetAmountTrans
			GRAND TOTAL	0.00

— Header data

— Detail data

The **header** section of the report returns just one row of data per key column value (the TenantHmy, in this case).

The **charge** section of the report returns as many rows as there are charges to the tenant during the period the user identifies at run time.

### Field Names

Because this example uses a custom SQL script, identifying the field names and aliases used in the script is trivial (you can simply review the script). If you are using a Voyager analytics source to return data, identifying field names is more difficult.

If you are working with Voyager analytics data sources, use the Yardi Excel Add-In to identify field names. For more information about the Add-In, see Chapter 8, "Yardi Excel Add-In for YSR."

In some cases you can also force YSR to generate a SQL error that reveals field names. For more information about this method, see "Identifying Field Names for Voyager Analytics Reports" on page 108.

### Excel Data Types

Format row 1 of your Excel report template with the **General** data type. Apply other data type formats, like date formatting, to rows 2 and following. For an illustration, see "Smart Markers and Excel Data Types" on page 203.



Failure to apply the **General** data type to row 1 can cause YSR reports to return no data.

### Formulas, Totals, and Pivot Charts with Excel

One of the advantages of using Excel templates with YSR is that you can take advantage of all functionality native to Excel. You can design pivot charts, use formulas, hide data in hidden rows or columns, and more.

You can also use smart marker parameters to control formatting options like grouping and row-skipping. For full treatment of these topics, see Chapter 7, "Working With Smart Markers and Excel."

## Word Template Example

This section provides an example of a Word report template that displays data from the report sections illustrated in the preceding topics.

This template uses data from all three sections of the report, all of which correspond to named SELECT sections in a custom SQL script. In place of Voyager data, the template uses merge fields as stand-ins.

<b>CREDIT NOTE</b>					
Tenant	«TenantName»		GST Reg no.	M9-0010744 A	
Attention	«BillingFirstName» «BillingLastName»		Invoice Date	«InvoiceDate»	
			Lease Start	«TenantFromDate»	
			Lease End	«TenantToDate»	
			Unit No	«Units»	
Property Name	«PropertyName»				
Tenant Code	«TenantCode»				
DUE DATE	DESCRIPTION	FROM	TO	AMOUNT	
«TableStart:Charge»«DUEDATE»	«ChargeDesc»	«ChargeDateFrom»	«ChargeDateTo»	«NetAmountTrans»«TableEnd:Charge»	
				Total	«SumTranAmount»

Word templates differ significantly from Excel templates, as follows:

- Voyager does not print to screen any YSR reports that use Word templates.
- You must add a top-level select statement to any YSR report that contains a Word template.
- Voyager does not apply the number and date formatting of the secured user when generating Word documents.

### Merge fields

- By default, Word merge fields return one row of data only. To return expanding row data, you must:
  - Add a merge field using the **TableStart** keyword, followed by the section code of the expanding row data section. For example: <<TableStart:charge>>
  - Add a merge field using the **TableEnd** keyword, followed by the section code of the expanding row data section. For example: <<TableEnd:charge>>

- In between the TableStart and TableEnd merge fields, add tokens. Tokens are merge fields with the format <<fieldname>>.



Merge fields do **not** use section codes (with the exception of the TableStart and TableEnd merge fields).

To map merge fields to report sections, therefore, Voyager looks first at the top-level SQL select statement for a matching column name or alias. If the top-level SQL select statement does not contain a match, Voyager looks at all underlying report sections.

- To add merge fields to a Word template, you must use Word's procedures for inserting merge fields. (You cannot simply type the greater/less than symbols (<</>>) around a field name.) For more information, see Word's online Help.

### Subtotals and formulas

Word templates, unlike Excel, have no built-in capacity for formulas or totals. If you want to provide a total, therefore, you must calculate the total inside of a SQL script and retrieve the total like any other data field. For example, the preceding template has a total line with this merge field: <<SumTranAmount>>. The <<SumTranAmount>> data comes from a SELECT section named **total** and scripted as follows:

```
//SELECT Total
Select
pr.hmy          TenantHmy
,sum(tr.stranamount) *-1    SumTranAmount

From person pr
inner join trans tr on tr.hperson = pr.hmy
and tr.hmy between 700000000 and 799999999
where 1=1 #conditions#
group by pr.hmy

//END SELECT
```

### Legacy form fields

As of Correspondence plug-in 7.12, YSR supports legacy form fields in Word templates and outputs the fields as completeable form fields on PDFs.

## Nested Tables in Word

With Correspondence Plug-in 7.5, you can add nested tables to Word-based report templates. Nested tables (tables within tables) help you display data in expanding blocks, rather than in expanding rows. Nested tables are especially useful for grouping, subtotaling, and formatting data in Word.

To illustrate, the following graphic shows a rendered report with lease charge data nested inside tenant blocks:

<b>Property: Sunrise Tower</b>		
Tenant: Advantage Mediation		
<table border="1"> <tr> <td>Monthly Rent: \$17,755.20</td> </tr> <tr> <td>Monthly CAM: \$1,380.96</td> </tr> </table>	Monthly Rent: \$17,755.20	Monthly CAM: \$1,380.96
Monthly Rent: \$17,755.20		
Monthly CAM: \$1,380.96		
Tenant: Beeker Group Architects		
<table border="1"> <tr> <td>Monthly Rent: \$39,445.92</td> </tr> <tr> <td>Monthly CAM: \$2,782.08</td> </tr> </table>	Monthly Rent: \$39,445.92	Monthly CAM: \$2,782.08
Monthly Rent: \$39,445.92		
Monthly CAM: \$2,782.08		
Tenant: Brian Lara Import/Export		
<table border="1"> <tr> <td>Monthly Rent: \$11,256.60</td> </tr> <tr> <td>Monthly CAM: \$863.52</td> </tr> </table>	Monthly Rent: \$11,256.60	Monthly CAM: \$863.52
Monthly Rent: \$11,256.60		
Monthly CAM: \$863.52		

In this example, the tenant block is itself nested within a block of property data. The template for this report is as follows:

```
«TableStart:prop»
Property: «PropName»
  «TableStart:Tenant» Tenant: «TenantName»
    «TableStart:ckd»
      Monthly Rent: «MonthlyRent»
      Monthly CAM: «MonthlyCam»
    «TableEnd:ckd»
  «TableEnd:Tenant»
«TableEnd:prop»
```

Each table displays data from a different report section and, in this case, a named SELECT section, as illustrated in the following mini script:

```
//SELECT Prop
select
  p.hmy                      MyProperty
  ,p.scode                     PropCode
  ,p.saddr1                    PropName
from property p
where 1=1 #condition1#
//END SELECT

//SELECT Tenant
select distinct
```

```

p.hmy                      PropHmy
,p.scode                    PropName
,t.slastname                TenantName
from property p
inner join tenant t on t.hproperty = p.hmy
inner join commkpidetail ckd on ckd.hTenant = t.hmyperson
where 1=1
and ckd.dtdate = '12-31-2008'
#condition1#
//END SELECT

//SELECT ckd
select
p.hmy                      PropID
,p.scode                    PropName
,t.SLASTNAME                TenantName
,'$'+Convert(Varchar(50),Convert(Money, Coalesce(sum(dleasemonth-
lyrent),0)),1) AS          MonthlyRent
,'$'+Convert(Varchar(50),Convert(Money, Coalesce(sum(dLeaseMonthly-
Cam), 0)),1) AS            MonthlyCam
from CommKPIDetail
inner join tenant t on CommKPIDetail.hTenant = t.HMYPERSON
inner join property p on t.HPROPERTY = p.hmy
where 1=1 and commkpidetail.dtdate = '12-31-2008' and p.hmy in (2)
group by t.SLASTNAME, p.hmy, p.scode
//END SELECT

```



For purposes of illustration, the key column (property.hmy) has different aliases in each SELECT section above.

You must relate each of the sub-sections to a parent section on the **Relations Setup** screen, accessible through the **Report Sections Setup** screen.

You relate a child section to a parent section by establishing a one-to-one relationship between the sub-section data and the parent data. Identify the aliases used in each data section to pick out the data column that uniquely identifies section data. In some cases, it is sufficient to establish one identity, as illustrated in the following graphic, where tenant data is linked to property data:

The screenshot shows two windows side-by-side. The left window is titled "Report Sections Setup" and lists three sections: Prop (Property), Tenant (Tenant), and ckd (charges). The right window is titled "Relations Setup" and shows a one-to-one relationship being established between the Tenant section and the Prop section.

**Report Sections Setup (Left Window):**

Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	Primary?
Prop	Property	Prop	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	MyProperty	<input checked="" type="checkbox"/>
Tenant	Tenant	Tenant	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	PropHmy	<input type="checkbox"/>
ckd	charges	ckd	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	PropID	<input type="checkbox"/>

**Relations Setup (Right Window):**

Relations Setup		
YSR Report:	nest test (nesttest)	
Report Code:	nest	
Section Code:	Tenant	
Parent Section:	Property(Prop)	
Parent Column Name	Child Column Name	Inactive?
MyProperty	PropHmy	<input type="checkbox"/>

Annotations below the windows:

- "Alias used in parent report section for a unique identifier" points to the "MyProperty" alias in the Parent Column Name column of the Relations Setup table.
- "Alias used in child report section for a unique identifier" points to the "PropHmy" alias in the Child Column Name column of the Relations Setup table.

In other cases it may be necessary to establish two identities, as for example when linking lease charge data to tenant and property data:

The screenshot shows the 'Report Sections Setup' dialog box. At the top, the report details are set to 'nest test (nesttest)' and 'nest'. Below this are 'Save', 'Close', and 'Help' buttons. The main area is titled 'Report Sections' and contains a table with columns: Section Code, Description, SELECT Name, Standard Analytics Report, Map Standard Analytics Filter, Tokens, Relations, Key Columns, and Primary?. The table rows are:

Prop	Property	Prop	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	Primary?
Tenant	Tenant	Tenant	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	PropHmy	<input checked="" type="checkbox"/>
ckd	charges	ckd	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	PropID	<input type="checkbox"/>

A modal window titled 'Relations Setup' is open over the table. It displays the same report details ('nest test (nesttest)', 'nest'). It also shows the 'Parent Section' dropdown set to 'Tenant(Tenant)'. Below this is another table:

Parent Column Name	Child Column Name	Inactive?
PropHmy	PropID	<input type="checkbox"/>
TenantName	TenantName	<input type="checkbox"/>

Annotations point from the 'PropHmy' and 'TenantName' columns to the text 'Aliases used in parent report sections for unique identifiers'. Annotations point from the 'PropID' and 'TenantName' columns to the text 'Aliases used in child report sections for unique identifiers'.

### Setup considerations for nested tables

When designing a report with nested tables, note that each table renders separately, and expands as many times as necessary to accommodate (1) the user's filtration and (2) the data returned by the corresponding data section.

Because each table renders separately, note that every table *and all its contents, including blocks of child data*, renders as many times as indicated by the resolution of the report at runtime. To illustrate, suppose you have a Property table and report section, and the underlying SELECT section includes a join to the tenant table. Even if the Property table in your template includes just the token for the property scode, the Property table renders as many times as there are tenants linked to that property.

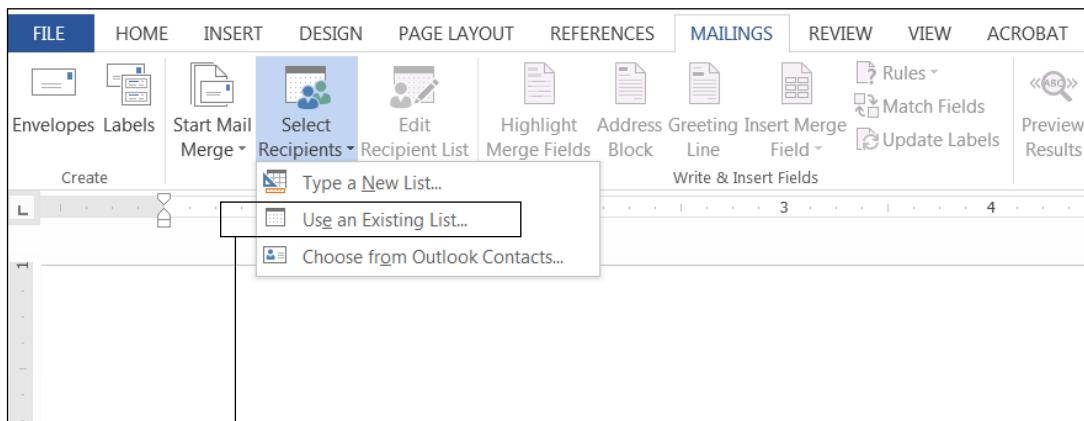
When scripting the data sections for each nested table, therefore, author SELECT sections carefully so that they retrieve only, and exactly, the data applicable to your report design.

# Working with Word 2013

This section discusses some special considerations for working with Word 2013.

## Adding Merge Fields with Word 2013

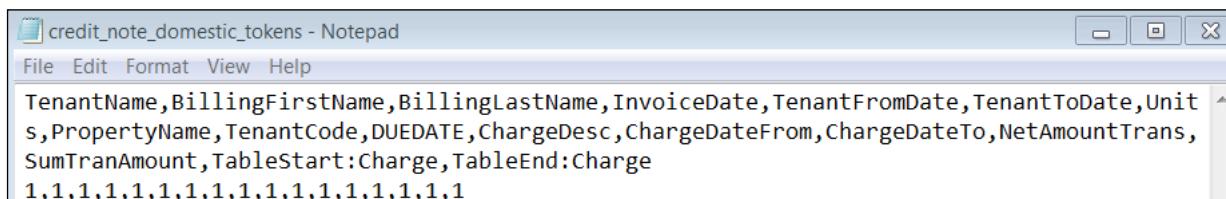
Up through Word 2010, you can add merge fields manually without identifying a data source for use in a mail merge. In Word 2013, by contrast, you must select a data source before you can insert merge fields.



## Option for adding a data source

To satisfy this requirement, you must create a token file containing the field names or aliases used in your YSR report. You can create the file manually, or you can use the Yardi Excel Add-in to build the file. For more information about the latter, see “Building Merge Token Files for Word Templates” on page 264.

The token file must contain at least one row of dummy data. Save the file as a CSV file. For example:



Then open Word, identify the token file as your data source, and you can add your tokens as merge fields to the template.



Word 2013 strips out the colon (:) in tokens like **TableStart:Charge**. Manually edit the token to replace the colon.



If necessary, you can remove a data source from a document by selecting **Start Mail Merge** (in the **Start Mail Merge** group) and selecting **Normal Word Document**.

## Formatting Dates in Word 2013 Templates

Because you have to provide Word 2013 with a merge field data source, there is no way to apply a format mask to date-type merge fields. Dates appear in long form (12/31/2015 12:00:00 AM).

There is no documented solution for date formatting Voyager analytics-based YSR reports in the culture of the user. If your report is based on a SQL script, however, you can modify your SQL script to convert dates when retrieving data. Use a convert clause like the following:

```
Select.... Convert(varchar(10),Table.DateField,101) DateAlias
```



This is a partial solution in that it applies one date format universally; it does not accommodate the local date formats of different users. There is no documented solution for date formatting Voyager analytics-based reports.

## PDF Template Example

This section provides an example of an PDF report template that displays data from the report sections illustrated in the preceding topics.

The screenshot shows a PDF template editor interface with the following elements:

- Header Area:** Contains a placeholder for the logo: `header.Property_Logo_af_image`.
- Text Fields:** Includes fields for `TenantName`, `TenantCode`, and `InvoiceNumber`.
- Billing Address Section:** Labeled `BILLINGADDRESS`.
- Print Options:** A dropdown menu for `TenantType` with options `Bill To Customer`, `Print`, and `No Print`. There is also a `BILLINGADDRESS` button.
- Report Footer:** Displays the text `BTW-nummer:`
- FACTUUR Section:** Lists fields for Factuurnummer, Factuurdatum, Vervaldatum, Ref. Nummer, Debiteurenkode, Perceel, and Adres, each followed by a colon and a merge field placeholder.
- ADRES EIGENAAR Section:** A label for the address section.
- Print Buttons:** A group of buttons labeled `Original`, `CusPrint`, `Copy`, and `Reprint`.

For more information, contact Yardi technical support.

## Report Output Example

The report sections and templates illustrated in the preceding topics generate the following report:

<b>CREDIT NOTE</b>				
Tenant	Five Santa Barbara Tenant	GST Reg no.	M9-0010744 A	
Attention	Jane Jetson	Invoice Date		
		Lease Start	01-01-2013	
		Lease End	31-12-2019	
		Unit No	5	
Property Name	Santa Barbara Property			
Tenant Code	t0000029			
<b>DUE DATE</b>	<b>DESCRIPTION</b>	<b>FROM</b>	<b>TO</b>	<b>AMOUNT</b>
01-01-2013	Rent	01-01-2013	31-01-2013	-1,250.00
01-01-2013	Parking	01-01-2013	31-01-2013	-125.00
01-01-2013	Miscellaneous Income	01-01-2013	31-01-2013	-125.00
			<b>Total</b>	-1500.00

## Report Section Setup

### In this section:

Setting up Report Sections.....	101
Adding Static Attachments to Report Sections.....	105

This section describes how to set up report sections.

### Setting up Report Sections

When you set up a report section, you provide basic configuration details that tell Voyager how to relate the report section data to your report templates. Most importantly, you must:

- Map the data source to a section code used in your report templates.
- Identify the key column of the data source.
- Indicate a primary report section (if there is no data retrieved by the primary section, Voyager skips the report).
- Indicate whether a section includes just one row of data or multiple rows of data.

### To set up a report section

- Select **Admin > YSR Correspondence > Setup Report**. The **YSR Correspondence Setup** screen appears.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see “Adding the YSR Admin Menu” on page 162.

- Complete the filter and click **Submit**. The **YSR Report Setup** screen appears.
- If it is not already active, click the **Report Setup** tab.
- Click the **Sections** button corresponding to the sub-report containing the sections you want to set up. The **Report Sections Setup** screen appears.

Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	Primary?	Multiple Rows?
header	header	header	▼	□	□	□	TenantHmy	<input checked="" type="checkbox"/>	□
charge	charges	DomChargeQuery	▼	□	□	□	TenantHmy	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
total	total	total	▼	□	□	□	TenantHmy	<input type="checkbox"/>	□
			▼	□	□	□		<input type="checkbox"/>	<input checked="" type="checkbox"/>
			▼	□	□	□		<input type="checkbox"/>	<input checked="" type="checkbox"/>
			▼	□	□	□		<input type="checkbox"/>	<input checked="" type="checkbox"/>

- Complete the fields. For more information, see “Report Sections Setup Screen Reference” on page 103.
- Click **Save**.
- Click **Close**.

## Report Sections Setup Screen Reference

The screenshot shows the 'Report Sections Setup' screen. At the top, there are fields for 'YSR Report:' (set to 'Credit Note Domestic (creditND)') and 'Report Code:' (set to 'cnd'). Below these are three buttons: 'Save', 'Close', and 'Help'. Below the buttons is a navigation bar with tabs: 'Report Sections' (which is selected) and 'Attachment Setup'. The main area contains a table with columns: Section Code, Description, SELECT Name, Standard Analytics Report, Map Standard Analytics Filter, Tokens, Relations, Key Columns, Primary?, Multiple Rows?, and Grouping Col. There are four rows in the table:

Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	Primary?	Multiple Rows?	Grouping Col
header	header	header	<input style="width: 100px; height: 20px;" type="button" value="..."/>	<input style="width: 100px; height: 20px;" type="button" value="..."/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
charge	charges	DomChargeQuery	<input style="width: 100px; height: 20px;" type="button" value="..."/>	<input style="width: 100px; height: 20px;" type="button" value="..."/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
total	total	total	<input style="width: 100px; height: 20px;" type="button" value="..."/>	<input style="width: 100px; height: 20px;" type="button" value="..."/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy	<input type="checkbox"/>	<input type="checkbox"/>	
			<input style="width: 100px; height: 20px;" type="button" value="..."/>	<input style="width: 100px; height: 20px;" type="button" value="..."/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	

**Section Code** The section code that appears in the Excel or Word template that corresponds to this report section.

**TIP** Excel templates use smart markers to represent Voyager data. Smart markers use this syntax: &=sectioncode.fieldname. Word templates refer explicitly to section codes only inside of tables. For more information about sections and how they related to Word templates, see “Word Template Example” on page 93.

**Description** For reference only.

**SELECT Name** (Applicable to report sections that use custom SQL scripts only.)

The name of the SELECT section that retrieves data for this report section. This field remains blank if the report section uses Voyager analytics to retrieve data.

**WARNING** Do not embed numeric characters into the body of the SELECT name anywhere except the end of the name. A name like “Select42” is valid, but “Select43A” might not function as expected.

**Standard Analytics Report** (Applicable to report sections that use Voyager analytics data sources.)

The Voyager analytics data source that retrieves data for this report section. This field remains blank if the report section uses a custom SQL script to retrieve data.

**Map Standard Analytics Filter** (Applicable to report sections that use Voyager analytics data sources.)

Opens the **Map Standard Analytics Filter** screen where you can map the filter conditions in the analytics report to the custom filter fields of your YSR report filter.

For more information about using Voyager analytics with your templates, see “Voyager Analytics Report Sections” on page 41 and “Adding Voyager Analytics Report Sections” on page 42.

<b>Tokens</b>	(Applicable to report sections that use some Voyager analytics data sources only.)  Opens the <b>Custom Token Setup</b> screen where you can add a custom token to a Voyager analytics report. Custom tokens represent additional columns of data.  <b>NOTE</b> The <b>Tokens</b> button  becomes active for a limited number of Voyager analytics. You can access this button only for analytics report with a public interface returning a select command (i.e., an interface with 'String' as the Return Type). You cannot add tokens to analytics reports that return a .NET type datatable.  For more information about custom tokens, see "To add a Voyager analytics report section" on page 43.
<b>Relations</b>	(Applicable for report sections that are for use in nested tables in Word report templates.)  Access point for the <b>Relations Setup</b> screen, where you can relate data from a child report section to a parent report section. For examples and more information, see "Nested Tables in Word" on page 94.
<b>Key Columns</b>	The primary key or unique identifier of the report section. Voyager uses this column to link data to other sub-report and section data.  <b>TIP</b> Most Voyager analytics data sources use 'propertyId' as the alias for the key column. Exceptions include Investment Management (investorId and investmentId) and Condo (OwnerHmy).  <b>TIP</b> If you want to email or attach reports, your report must use a numerical identifier (hmy) rather than a string (scode) as its key column.  <b>NOTE</b> As a best practice, do not use 'property' as the alias for the key column. Key columns aliased 'property' can cause reports to display no data for users with property list security. This applies to all aliases for all key columns in the report.
<b>Primary</b>	Specify at least one section as primary. If the primary sections do not return any data for any key column value, then Voyager skips the sub-report for that value.  <b>TIP</b> If you are creating a Word template with header and detail sections, select the <b>Primary</b> check box for both sections.
<b>Multiple Rows</b>	(Voyager selects this check box by default for new sections.)  Indicates that the section includes multiple rows of data.  <b>NOTE</b> If Voyager returns multiple rows of data for this section but you do not select this check box, Voyager prints only the first value.  <b>TIP</b> If you are configuring header data, or other data that must appear just once, clear this check box.
<b>Grouping Columns</b>	This field works in conjunction with the <b>Summary Columns</b> field.  Specify a list of columns, separated by commas. Voyager returns distinct records for each column.  <b>NOTE</b> Use this option only if you cannot apply grouping and summarizing in your Excel template.
<b>Summary Columns</b>	This field works in conjunction with the <b>Grouping Columns</b> field.  Specify a list of columns, separated by commas. Voyager sums the column values to create distinct records for the <b>Grouping Columns</b> field.  <b>NOTE</b> Use this option only if you cannot apply grouping and summarizing in your Excel template.
<b>Inactive</b>	Excludes the section from use.

## Adding Static Attachments to Report Sections

You can add static attachments to YSR reports. For example, you might want to add the same legal notice or other disclosure statement to all reports. Alternatively, you might want to attach different documents depending a user's selection in a custom filter field at run time. YSR can accommodate both scenarios.



Static attachments must be of the same file type as your report template. That is, you can add Excel documents to Excel-based YSR reports and Word documents to Word-based reports. You cannot attach PDFs.

Attachments are not supported when publishing reports to screen.

You add attachments to report sections by adding a SELECT statement to the **Attachment Setup** tab on the **Report Sections Setup** screen.

The screenshot shows the 'Report Sections Setup' dialog box. At the top, there are fields for 'YSR Report:' (Lease Activity History (LeActHis)) and 'Report Code:' (ActHis). Below these are 'Save', 'Close', and 'Help' buttons. At the bottom, there are tabs for 'Report Sections' (selected) and 'Attachment Setup'. The 'Attachment Select:' section contains the following SQL code:

```
Select p.scode PropCode, pd.sFile Filepath
From PMDocs pd
Inner Join property p on p.hmy = pd.hRecord
Inner Join AttachmentType att on att.hMy = pd.hAttachmentType
Inner Join Owner o on o.hMyPerson = p.hLegalEntity
Where pd.iType=3 and att.sDesc = 'Retailer Application Form'
#Condition01#
```

If you use Orion for SharePoint, you can also attach an Orion document to your report.

The next sections of this document describe the requirements for the SELECT statements you can use in each scenario and provide examples.

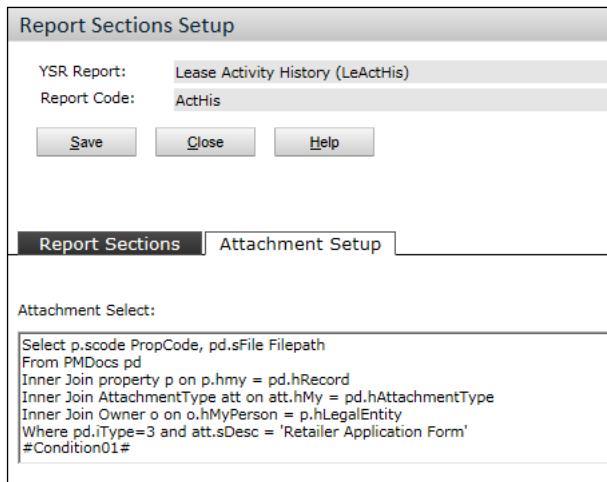
### This section includes the following procedures:

- |   |     |
|---|-----|
| To add an attachment to a sub-report .....                      | 105 |
| To add different attachments depending on user's selection..... | 106 |
| To attach an Orion document to sub-reports.....                 | 107 |

### To add an attachment to a sub-report

You can add additional files to a sub-report by using a select statement that retrieves, from the data table you specify (typically the PMDocs table), the complete file path of the document you want to attach. Enter the select statement on the **Attachment Setup** tab on the **Report Sections Setup** screen corresponding to the sub-report to which you want to attach the file.

For example, the following graphic shows a select statement that attaches a document to the OpenInv sub-report. In this example, the complete file path to the document is contained in the sFile column of the PMDocs table. You must alias the column using the keyword FilePath.



The select statement must obey the following constraints:

- The select statement must retrieve both the key column and the column containing the complete file path of the document you want to attach.
- The key column must have the same alias as specified in the **Key Column** field on the **YSR Report Setup** screen.
- The column containing the file path must have the alias **FilePath**.
- The FilePath column must contain the complete file path of the document.
- The file type extension of the document you want to attach must be exactly the same as the file type extension of the template in the sub-report.

#### To add different attachments depending on user's selection

You can design your YSR report to include different attachments depending on a user's selection at run time. For example, you might create a YSR report that includes different attachments depending on whether the report concerns rent charges or common area maintenance (CAM) charges. You can accomplish this by the following:

- 1 Creating a custom filter field where users can select rent or cam
- 2 Using a CASE expression to select one attachment type for rent charges and another for CAM charges.
- 3 Use the attachment type to identify the file path to the document you want to attach.

Because Voyager stores attachment types in a different table (the AttachmentType table) than the table containing file paths to correspondence documents (the PMDocs table), you must use a JOIN clause to link the two.

The select statement must obey the following constraints:

- The select statement must retrieve both the key column and the column containing the complete file path of the document you want to attach.
- The key column must have the same alias as specified in the **Key Column** field on the **YSR Report Setup** screen.
- The column containing the file path must have the alias **FilePath**.
- The FilePath column must contain the complete file path of the document.
- The file type extension of the document you want to attach must be exactly the same as the file type extension of the template in the sub-report.

The following graphic provides an example of a script that retrieves a different attachment depending on whether the user selects rent or cam in a custom filter field (#ChargeType#).

```

File Edit Format View Help
SELECT Distinct
    isnull((CASE len(tc.sInvNum)      WHEN 0 THEN NULL ELSE tc.sInvNum END
), tc.hParent2) InvoiceNumber
, pd.sFile FilePath
, p.sCode PropCode
FROM person pr
    INNER JOIN trans tc ON (
        tc.hperson = pr.hmy
        AND tc.iType = 7
        AND tc.hmy BETWEEN 700000000
        AND 799999999
    )
    INNER JOIN Property p ON p.hmy = tc.hProp
    Inner join pmdocs pd on pd.hRecord = p.hmy and pd.iType = 3
    Inner Join AttachmentType at on at.hMy = pd.hAttachmentType
        WHERE tc.iType = 7
        AND isnull(tc.isubtype, 0) <> 1
        AND isnull(tc.hParent4, 0) = 0
        AND upper(IsNull(tc.suserdefined2, ' ')) NOT LIKE ':CI%'
        AND isnull(tc.hParent3, 0) = 0
        AND isnull(tc void, 0) = 0
        and pd.iType=3
    and at.sDesc = (Case '#ChargeType#' When 'rent' THEN 'Rent_Terms' WHEN 'cam'
THEN 'CAM_Terms' Else '' End)
#Condition02#
#Condition03#
#Condition04#
#Condition05#
#Condition06#
#Condition07#
#Condition08#
#Condition09#

```

### To attach an Orion document to sub-reports

You can attach an Orion document to a sub-report by writing a select statement that retrieves, from the PMDocs table, the key column of the YSR report setup and the hmy of the PMDocs table. The select statement must use two keywords: **KeyColumn** and **PMDocID**.

Example select statement:

```

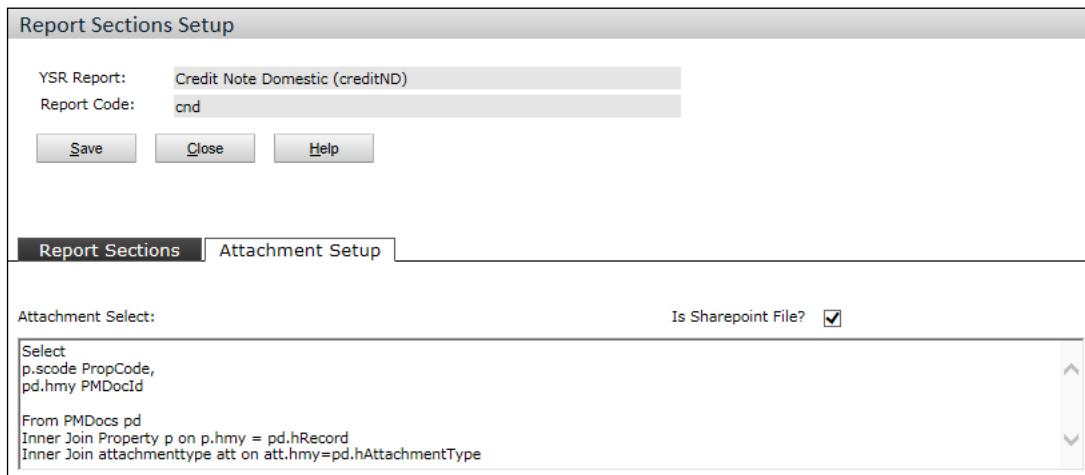
File Edit Format View Help
Select
p.sCode PropCode,
pd.hmy PMDocID

From PMDocs pd
Inner Join Property p On p.hMY=pd.hRecord
Inner Join AttachmentType att On att.hMy=pd.hAttachmentType
Inner Join Owner o On o.hMyPerson=p.hLegalEntity

Where pd.iType=3 And att.sDesc = 'Retailer Application Form' #Condition01# #Condition02#

```

On the **Attachment Setup** tab, you must also select the **Is Sharepoint** check box.



## Report Template Setup

### In this section:

Identifying Field Names for Voyager Analytics Reports .....	108
Adding Images to Word Templates .....	111
Displaying Text Containing HTML Tags in Word Templates.....	112
Downloading YSR Report Templates.....	113

This section describes additional report template setup tasks.

### Identifying Field Names for Voyager Analytics Reports

Report templates contain the text and formatting of your report documents. They also contain place-holders for Voyager data, which YSR retrieves at run time. Excel templates use smart markers, and Word templates use merge fields, to stand in for Voyager data.

In order to create smart markers and merge fields, you must know the field names used in the underlying data sources of your report. If you are working with a custom SQL script, this is trivial (you can simply review the script). When working with Voyager analytics data sources, however, you may not have easy access to the field names (or aliases) used in the data source.

You can access the field names in Voyager analytics data sources as follows:

- You can download and install the Yardi Excel Add-In (for use in Excel only). The Yardi Excel Add-In contains the section codes and field names of the Voyager analytics available in YSR. For more information, see Chapter 8, "Yardi Excel Add-In for YSR."
- You can access the field aliases used in most Voyager analytics by forcing YSR to generate a SQL error. The error message reveals the field aliases. This method is discussed below.



This method works only for Voyager analytics reports that return a String type. For those that return a .NET type data table, the **Tokens** button is disabled.

### To identify the field aliases used in Voyager analytics

- 1 Select **Admin > YSR Correspondence > Setup Report**. The **YSR Correspondence Setup** screen appears.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see "Adding the YSR Admin Menu" on page 162.

- 2 Complete the filter and click **Submit**. The **YSR Report Setup** screen appears.
- 3 On the **Report Setup** tab, click the **Sections** button corresponding to the sub-report that uses the Voyager analytics report you want to review. The **Report Sections Setup** screen appears.

Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	Primary?
AP	Expense Distribution		AP Analytics	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>
				<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
				<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

- 4 Click the **Tokens** button  corresponding to the Voyager analytics field aliases you want to review. The **Custom Token Setup** screen appears.

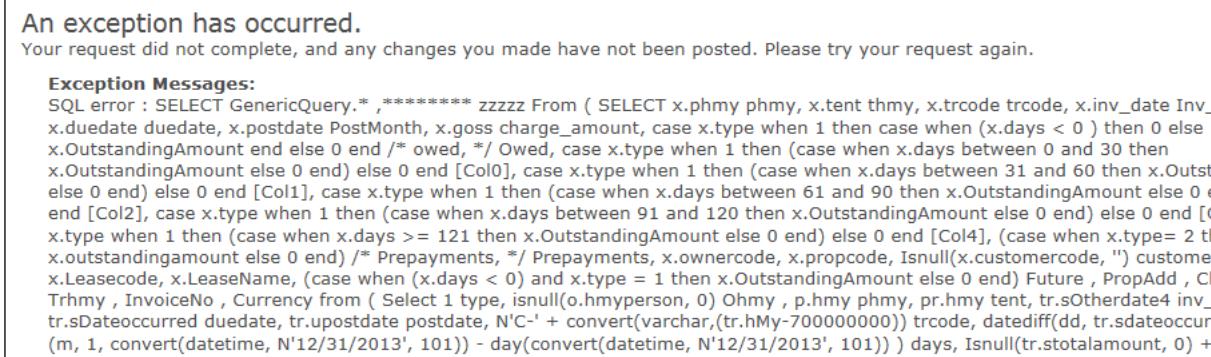
- 5** Enter an invalid token that generates a SQL error, as illustrated in the following graphic.

Token Name	Select Query	Inactive?
zzzzz	*****	<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

- 6** Click **Save**.

- 7** Click **Close**.

- 8** Close the setup screens and generate the report by selecting **Admin > YSR Correspondence > Generate Report** and choosing the report containing the invalid token. Voyager generates a SQL error.



The error message contains all the field aliases necessary for completing your report templates.

**NOTE** Make sure that the invalid token is the only erroneous part of your YSR report setup. Otherwise, Voyager may display other error messages that do not reveal the field aliases.

- 9** For convenience, use a tool to format the SQL.



There are many free tools available. Search the Internet for SQL formatter.

This step is not strictly necessary, but it makes it easier to see the field aliases.

- 10** Use the field aliases in your report template smart markers or merge fields as necessary.

## Adding Images to Word Templates

To add images to Word templates, use the following syntax in your Word merge fields:

```
<<image:ImageAlias>>
```

For example, suppose your report includes a SQL section that retrieves an image file name and aliases it **tenantimage**. To display the image in your Word template, add this merge field:

```
<<image:tenantimage>>
```

### Controlling Image Size

One way to control image size in Word is to place the image inside a table cell. Size the cell as desired and use Word's table cell properties to fit the contents to the size of the cell.

If you do not fit the contents to the cell, Voyager crops the image.

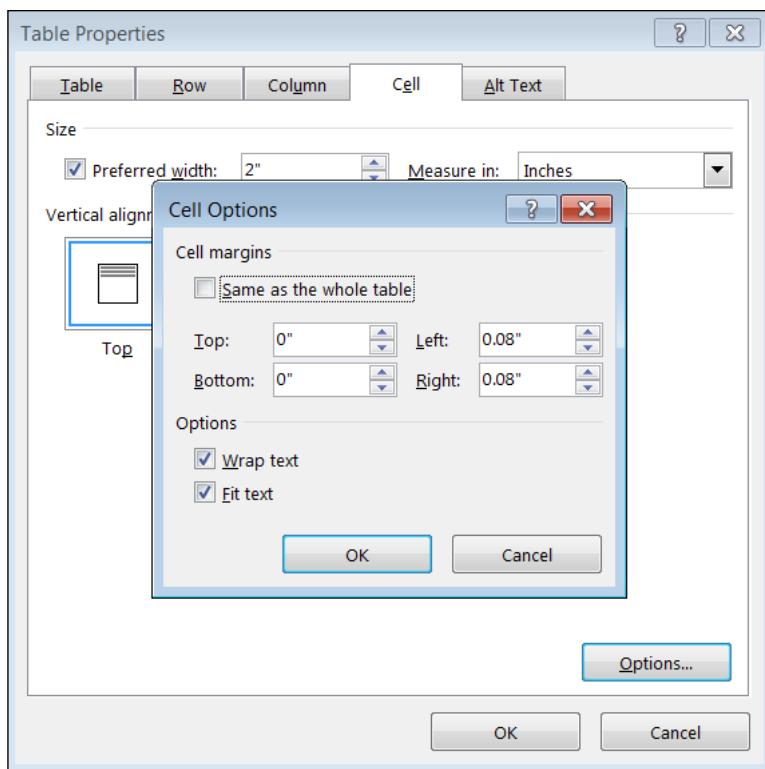
#### To fit contents to a cell



This section describes how to edit the **Fit text** property of a table cell in Word 2013. In prior versions of Word, edit the **Fit to Cell** property of a table cell.

- 1 Select the table cell you want to manipulate.
- 2 On the ribbon, in the **Table Tools** area, select **Layout**.
- 3 Open the **Cell Size** group. The **Table Properties** screen appears.
- 4 Click the **Cell** tab.
- 5 Click **Options**.

- 6** Select the **Fit text** check box.



- 7** Click **Ok**.

- 8** Click **Ok**.

## Displaying Text Containing HTML Tags in Word Templates

You can display HTML-tagged text in Word templates.

This is useful if you want to retrieve and display HTML-formatted text stored in the MEMO table or other notes table. For example, the following graphic shows HTML-formatted data in the stext column of the MEMO table:

	HMY	HFILERCD	IFIELD	IFILETYPE	UDA...	ITYPE	HSTAT...	STEXT	DTE...	HPR...
1	590	22	2	91	NULL	NULL	NULL	<b>Art Design and Construction</b> for interior ...	NULL	NULL

To preserve the text formatting, alias the data column containing the text with the prefix **html**. For example, you might retrieve the text data pictured above with the following Select statement:

```
//Select Notes
Select stext [htmlstext] --alias with html prefix
from memo
where #condition1#
//End Select
```

Then use the alias in your Word template merge fields as normal:

<<htmlstext>>

YSR displays the text as formatted with HTML tags.

## Downloading YSR Report Templates

You can download Word or Excel templates for any of the YSR reports in your database.

### To download a YSR Report template

- From the **YSR Admin** side menu, select **Setup > YSR Administration**. The **YSR Administration** screen appears.



For information about the **YSR Admin** side menu, see “Adding the YSR Admin Menu” on page 162.

- Click the **Download YSR Supporting Files as Zip** tab.

The screenshot shows the YSR Administration interface. At the top, there's a navigation bar with tabs: Inactive YSR Reports, Menu Items with YSR Report Issues, and a highlighted tab labeled "Download YSR Supporting Files as Zip". Below the tabs, there's a section titled "Download Files for:" with a dropdown menu set to "Lease Activity History (LeActHis)". There's also a checked checkbox for "Delete older zips for this YSR report (recommended)". At the bottom left is a "Save" button.

- Select a report from the **Download Files for** field.



The **Download Files for** field includes both active and inactive reports.

- 4 Click **Save**. If the operation is successful, a .zip file link appears:

[Download zip of YSR files for Lease Activity History \(LeActHis\)](#)

- 5 Follow the prompts to save the file..



If the operation is only partially successful (usually due to missing files), an ErrorLog.txt link appears instead. Click the link to view the error log.

## Dynamic Template Selection

### In this section:

Selecting Templates According to Criteria in Top-Level Select.....	115
Selecting Templates According to User's Filter Criteria.....	116
Selecting Templates Using Other Runtime Options.....	118

There are multiple ways that you can select a report template dynamically at runtime. You can:

- Select report templates based on criteria established in your top-level select statement.

For example, you can generate reports using one template for some properties and another template for other properties. This method requires a top-level select statement and uses a token, structured like a smart marker (#HeaderFooter.FieldAlias#), in place of the report template file name on the **Report Setup** tab.

- Select report templates based on user's criteria.

This method uses a SELECT statement in place of the report template file name on the **Report Setup** tab. The SELECT statement includes a token that corresponds to a top-level filter field that the user completes at runtime.



You can accommodate user's criteria using the first of the two methods listed above, as well. The first method requires a top-level select statement, however, so the second method is documented for use with reports that have no top-level select statement. The second method may also prove adaptable for uses not considered here.

## Selecting Templates According to Criteria in Top-Level Select

By adding a CASE statement to your top-level select, you can select templates dynamically based on criteria you establish in advance of the user's input. For example, you can use one report template for some properties or owners and another template for other properties or owners.

This approach relies on the following special-purpose token:

#HeaderFooter.FieldAlias#

This token stands in for the report template filename (excluding the file type), as it appears in the **Template File** column on the **Report Setup** tab.

Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?
10	RentRM	#HeaderFooter.TempName#.xlsx						

In this illustration, **TempName** is the alias of the column that returns a report template filename. For example, the following script uses a CASE statement to dynamically retrieve different filenames depending on property codes:

```

SELECT
p.hmy PropId
,p.scode PropertyCode
,case when (p.scode in ('nlfund')) then 'NLFundTemplate' when (p.scode
in ('cnoff01')) then 'CNOFF01Template'
else 'DefaultTemplate' end TempName

FROM property p
WHERE p.hmy in (#propertylist#)

```

Suppose a user then generates the report for properties comoff01, cnoff01, and nlfund. The following graphic shows the results, with filenames generated dynamically in the last column:

	PropId	PropertyCo...	TempName
1	2	comoff01	DefaultTemplate
2	108	cnoff01	CNOFF01Template
3	138	nlfund	NLFundTemplate

Voyager then generates the report using the different report templates for each property.

## Selecting Templates According to User's Filter Criteria

You can provide users with an option to select from multiple templates when they generate the report. For example, you might create two Trial Balance templates, one for use in Germany and one for use in the UK.

To accommodate this situation, create two templates for use with the same report data. Then, create a custom filter field in the YSR report filter where users can identify the template they want to use. Last, in place of the template file name, use a select statement on the **YSR Report Setup** screen to complete the **Template File** field.

On report generation, Voyager selects the template dynamically, based on the user's input.

### Setup Steps

1 Create your report templates and save them to the Reports path using the same root filename plus an appendix to differentiate them. For example:

- YSR\_TrialBalance\_de.xlsx
- YSR\_TrialBalance\_uk.xlsx



The templates must have the same file type.

2 Create a custom filter field of the **List** type where users can select a template. For example, the fourth filter in the graphic gives users the option to choose between the German and UK templates.

Report Filters Setup																																																											
Merged Report: Trial Balance (TrialBal)																																																											
<input type="button" value="Save"/>		<input type="button" value="Close"/>																																																									
<table border="1"> <thead> <tr> <th></th> <th>Field Name</th> <th>Label</th> <th>Sequence</th> <th>Type</th> <th>Lookup Name</th> <th>Multi Select?</th> <th>Parent</th> <th>Mandatory?</th> <th>List Values</th> </tr> </thead> <tbody> <tr> <td></td> <td>PropertyCode</td> <td>Property</td> <td>1</td> <td>Lookup List</td> <td>ysiPropertyOrListLookup</td> <td><input checked="" type="checkbox"/></td> <td></td> <td><input checked="" type="checkbox"/></td> <td></td> </tr> <tr> <td></td> <td>Book</td> <td>Book</td> <td>2</td> <td>Lookup List</td> <td>ysiBookLookup</td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td></td> <td>GLAccounts</td> <td>GLAccounts</td> <td>3</td> <td>Lookup List</td> <td>ysiAccountLookup</td> <td><input checked="" type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> <td></td> </tr> <tr> <td></td> <td>SelCountry</td> <td>Select Country</td> <td>4</td> <td>List</td> <td></td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> <td>^de^uk</td> </tr> </tbody> </table>											Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	List Values		PropertyCode	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>			Book	Book	2	Lookup List	ysiBookLookup	<input type="checkbox"/>		<input type="checkbox"/>			GLAccounts	GLAccounts	3	Lookup List	ysiAccountLookup	<input checked="" type="checkbox"/>		<input type="checkbox"/>			SelCountry	Select Country	4	List		<input type="checkbox"/>		<input type="checkbox"/>	^de^uk
	Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	List Values																																																		
	PropertyCode	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>																																																			
	Book	Book	2	Lookup List	ysiBookLookup	<input type="checkbox"/>		<input type="checkbox"/>																																																			
	GLAccounts	GLAccounts	3	Lookup List	ysiAccountLookup	<input checked="" type="checkbox"/>		<input type="checkbox"/>																																																			
	SelCountry	Select Country	4	List		<input type="checkbox"/>		<input type="checkbox"/>	^de^uk																																																		



Take note of the strings you enter in the **Field Name** field and the **List Values** field. You must refer to these in the next step.

- 3 On the **YSR Report Setup** screen, enter a select statement in the **Template File** field. Use a select statement like this:

```
Bå de
when 'uk' then '_uk' else '' End) + '.xlsx'
```



The select statement refers to the root of the report template filenames (YSR\_TrialBalance), the field name of the custom filter field (SelCountry), and the list values that appear to users (de and uk). The select statement instructs Voyager to use the report template with the appropriate appendix (\_de or \_uk) and file type (.xlsx).



The select statement must include the keyword SELECT and must use the filter mapping convention of using a hash mark (#) to surround the name of the defined filter element. Without the hash-marked token, Voyager does not recognize the string as a SELECT statement.

The select statement can have 1,000 characters maximum.

The screenshot shows the 'YSR Report Setup' dialog box. In the 'Code' field, 'TrialBal' is entered. In the 'Description' field, 'Trial Balance' is entered. Under 'Top-Level Select Statement', the following SQL code is present:

```
select p.hmy propertyId,p.scode sCode from property p
where
#SelCountry# = 'uk'
OR Script File
```

In the 'Key Column' field, 'propertyId' is selected. The 'Inactive' checkbox is unchecked. In the 'Notes' panel, the following details are listed:

- FinType : TRIAL BALANCE
- SuppressZero : 1
- TreeLevel : 2
- Filters : PropertyCode,TreeCode,FromPeriod,ToBook, TreeCode, IsConsolidate

At the bottom of the dialog, there are buttons for Save, New, Save Copy, Define Filters, Map Top-Level Filter, Dump SQL, Delete Setup, and Help. Below the dialog, a navigation bar includes 'Report Setup', 'Attachment & Email', 'Additional Attachment & Email', 'Output Options', and 'Digital Signature'. A table below the navigation bar lists report configurations:

Order	Report Code	Template File	Script File	Page Break Column	Map Filter	Sections
10	TrialBal	Select 'YSR_TrialBalance' + (Case '#SelCountry#')		propertyId		

**Select statement for choosing between multiple templates**

When users generate the Trial Balance report, they can choose the country (de or uk). Voyager selects the template dynamically, depending on the user's choice.

## Selecting Templates Using Other Runtime Options

This section suggests other possible methods for using a script, in place of the report template file name, to dynamically select a report template based on runtime options. With an appropriately crafted script, you can retrieve a different report template depending on:

- Attachment or Email options
- Publishing options (to Portal, to SharePoint)
- Filter field value
- User ID

These data points are captured in slightly different ways in session-related tokens, in the iFilter table, and in the YSRReportHistory table. Each set of data can be leveraged in a SELECT statement supplied in the **Template File** field in place of a template filename, as illustrated here:

A suitable select statement must:

- Contain 1,000 characters maximum.
- Include a token, indicated by hash marks (#...#). Without the token, Voyager does not recognize the string as a Select statement. If you have no real need for the token, write a tautology like when

```
SELECT case when '#property#' = '#property#' and '#@@OUTPUTTYPE#' = 'screen' then 'ARTest.xlsx' when '#@@OUTPUTTYPE#' = 'Excel' then 'ARTestExcel.xlsx' end
```

**The iFilter table** captures all report parameters, including publication options (OutputType, IsGrid, et.c). Whenever a user generates a YSR report, Voyager inserts a record (or overwrites an existing record, if the user has generated the report previously) into the iFilter table before generating the report. For example, the following graphic shows the iFilter records for three reports generated by the DBO (hence 'blank' in the sUserID column and '\_0\_' in the sFilterID column).

	SFILTERID	SUSERID	SURL
1	CustomCorrespondence_0_118	blank	&Owner=nlown&Property=&DateFrom=&DateTo=&MergedReportId=118&OutputType=
2	CustomCorrespondence_0_166	blank	&Property=comoff01&Building=&MergedReportId=166&OutputType=Screen&IsGrid=
3	CustomCorrespondence_0_167	blank	&property=resnc01&MergedReportId=167&OutputType=Screen&IsGrid=0&IsAttach=

To use data in the iFilter table, write a select statement that responds to parameters supplied in the SURL column. Data is identified by the string in the sFilterID column with this format:

CustomCorrespondence\_UserID\_YSRReportHmy

**The YSRReportHistory table** captures similar data (all filter parameters, but no publication options), and organizes data by date and time. Each successive report generates a new record in the table, so you can find the most recent record by retrieving the record with the maximum hmy. The following graphic illustrates partial data in the YSRReportHistory table (sMessage and bComplete columns not shown).

	hMy	sCode	hUs...	dtStart	sParms	dtLastUpdate
1	1	ASrTest	0	2018-04-12 15:49:29.303	&property=31	2018-04-12 15:49:33.9
2	2	ASrTest	0	2018-04-12 15:49:47.203	&property=31	2018-04-12 15:49:47.2
3	3	RentRoll	0	2018-04-12 15:51:12.353	&PropertyList=2&FromDate=2008-12-31&Building=&Ch...	2018-04-12 15:51:13.6
4	4	RentRoll	0	2018-04-12 15:52:46.983	&PropertyList=2&FromDate=2008-12-31&Building=&Ch...	2018-04-12 15:52:47.0
5	5	RentRoll	0	2018-04-12 15:52:58.403	&PropertyList=2, 108&FromDate=2008-12-31&Building...	2018-04-12 15:52:58.4
6	6	ASrTest	0	2018-04-12 15:54:14.053	&property=31	2018-04-12 15:54:14.0
7	7	RentRoll	0	2018-04-12 15:54:59.773	&PropertyList=2, 108&FromDate=2008-12-31&Building...	2018-04-12 15:54:59.7



To write to the YSRReportHistory table, load the following package:

Opt\_ss\_YSR\_EnableReportHistory.pkg

The bComplete column returns -1 for successful report generation, 0 for a report that fails without generating an error message, and -2 for a report that fails with an error message. Error messages, if available, are stored in the sMessage column.

## CHAPTER 5

# Production and Delivery

### In this chapter:

Email, Attachments, and Publication.....	120
Output Options.....	134
Digital Signatures.....	141
Task Runner .....	148
Report Scheduler .....	151
Side Menus.....	158
Custom Help .....	163

This section describes options for producing and delivering YSR reports.

## Email, Attachments, and Publication

### In this section:

Attaching Reports .....	122
Attaching and Emailing Reports.....	124
Publishing to SharePoint.....	129
Publishing to Portal.....	131
Holding Emails .....	131
Static PDF Attachment Error Logs .....	132
Monitoring Reports .....	132
Reviewing YSR Report Generation History .....	133

This section shows how to attach reports to Voyager objects, email reports, publish reports to SharePoint and Portal, and monitor report activity.

YSR provides a robust publishing apparatus with many options for attaching, emailing, previewing, reviewing, and troubleshooting reports. With YSR, you can:

- Attach reports to Voyager objects.

- Email reports to the contacts associated with one or more Voyager objects.
- Email reports to recipients using email IDs.
- Preview reports before sending them.
- View sent emails.
- Manage reports for recipients without email.
- Bypass setup steps and send a report directly to one or more email address.

### Setup overview

To match reports with their object, Voyager relies on primary keys. That is, Voyager matches each rendered report with a corresponding Voyager object, contact, or email ID according to their key column value. Therefore, to email and attach reports, your report must use a top-level select statement and a numerical key column like `property.hmy` or `tenant.hmyperson`. (You cannot use string values like `property scodes` as the key column if you want to email or attach reports.)

Email and attachment go hand-in-hand: if you want to email reports, you must also attach them to Voyager objects. You can, however, attach reports to Voyager objects without emailing them.



If you want to email a report directly to one or more known email addresses (for testing, for example), you do not need a top-level select statement or key column. You can enter the address or addresses, separated by a semi-colon (;), directly on **Attachment & Email** screen, and Voyager sends reports directly to those addresses.

### User permissions

You can set user permissions for attaching and emailing reports. The relevant tokens are:

- Correspondence:YSR - Attach Report
- Correspondence:YSR - Email report

## Attaching Reports

You can set up YSR reports so that users have the option of attaching them to Voyager objects. Setting up attachments is required in order to email reports. To set up attachments, your report must have a numerical key column like property.hmy, and you must identify the key column by its alias on the **Attachments & Email** tab.

For example, the following graphic shows a YSR report that is set up to attach reports to property objects. **Propertyid** is the alias that the top-level select statement uses for property.hmy.

The screenshot shows the YSR Report Setup interface. On the left, there's a main configuration area with fields for Code (CommCh), Description (Commercial Charges), Top-Level Select Statement (a SQL query selecting propertyid and p.scode from property p where p.hmy is in (#PropertyCode#)), Key Column (propertyid), and Inactive (unchecked). Below this is a 'Save' button. On the right, there's a 'Notes' section with a large text area. A 'Attachment & Email' tab is open, showing the YSR Report (Commercial Charges (CommCh)), an unchecked Inactive checkbox, and three buttons: Save, Close, and Delete. Under 'Object Type Setup', the Object Type is set to 'Property' and the ID Column is 'propertyid'. The 'Attachment Setup' tab has 'Attach' checked and 'Email' unchecked. The 'Email Setup' tab includes fields for Attachment Type (a dropdown menu), Attachment Filename Column (Property\_Filename), Email Template (a dropdown menu), Email Type (a dropdown menu), Email To (Role) (a dropdown menu), and an unchecked 'Include All Contacts' checkbox. The 'Additional Recipients' section contains a table with columns for Object Type, ID Column, Email Type, Role, To/Cc/Bcc, and Include All Contacts. There are three rows in the table, each with dropdown menus for the first four columns and checkboxes for the last two.



The value in the **ID Column** field on the **Attachment & Email** screen must match the value in the **Key Column** field on the top part of the screen.

### Attachment file names

Define file name columns in the top-level select statement. For example:

```
, 'Lease Info' + o.uCode Owner_FileName
, 'Lease Info' + t.sCode Tenant_FileName
```

Enter the column in the Attachment File Name column on the Attachment & Email screen for each attachment type. YSR generates a unique file name for each attachment using these settings.

## Attaching reports to objects

Once you have completed the necessary setup steps, users can select the **Attach Reports** check box when generating the report:

Voyager generates the report (or reports, if generated for multiple entities) and attaches the report to each object.

Type	Description	Date	Attachment	Secure	Show on Portal	Detach
	CommCh_2_0	12/31/2008	CommCh_2_0.xlsx	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## Attachment file size

Voyager by default limits attachments to files less than 5MB in size. Administrators can override this restriction by loading the following package: optAttachmentNoFileSizeLimit.pkg.

For more information about contacts, roles, and email templates, see the *Voyager Workflows and Notifications User's Guide*.

For step-by-step procedures for setting up attachments, see "To set up attachment and email" on page 125.

## Attaching and Emailing Reports

This section shows how to set up email and attachment for YSR reports. Attachment is a precondition for email; you must attach reports to Voyager objects in order to email them.

There are many ways to configure email. You can:

- Enter an email address manually when setting up the report (useful for test reports).
- Send the report to the email ID associated directly with a person record (a tenant, employee, owner, vendor, or other person type record).
- Send the report to one or more contacts associated with a person record. (All contacts have roles; send the email to the contact with the appropriate role.) With this method, you can assign recipients to the **To**, **CC**, and **BCC** fields of the outgoing email.
- Send the report to multiple contacts associated with different Voyager objects. For example, you can configure a report to send a rent schedule to property contacts and to owner contacts.
- Require or prevent email and attachment; let users choose whether to attach and email.
- Set up report previews for first-time reports.

### Email IDs versus contacts

The following graphic illustrates the difference between email IDs and contacts:

The screenshot shows the 'Legal Entity' screen in Yardi Voyager. At the top, there is a 'Data' dropdown menu. Below it, the entity details are listed: Code (cnownon), Name (Ontario Corporation Ltd.), Address (9899 Airport Road), City (Mississauga), Prov-Post Code (ON L6T 2H7), Country (cn), and Notes (Ontario Corporation owns real estate in Ontario and manages these portfolios). To the right of these details is a table with columns for Email Address and Alternate Email, with 'ontcorp@ontcorp.ca' listed under Email Address. Below this table is another table with columns for Office and Home, with 'Home' listed. At the bottom of the screen, there are tabs for Properties, Tax Info, Other Info, Payment Info, and Contacts. The Contacts tab is currently selected. Below the tabs is a toolbar with buttons for Edit, New, Close, and Help. At the very bottom, there is a navigation bar with icons for Home, Back, Forward, and Search, followed by a link to 'Yardi Voyager Yardi Spreadsheet Reporting User's Guide'.

Role	Primary	Company	Name	Office	Home	Email
Executive	Y	Ontario Corp. Ltd.	Demetrios Costandinos	(905) 231-3133		Demetrios@OntarioCorp.Ltd.com
Billing Contact		Ontario Corporation Ltd.				ontcorp@ontcorp.ca

**Email ID of owner**: Points to the 'Email Address' field in the top-right table, containing 'ontcorp@ontcorp.ca'.

**Emails of owner's contacts**: Points to the 'Email' column in the bottom table, which contains two entries: 'Demetrios@OntarioCorp.Ltd.com' and 'ontcorp@ontcorp.ca'.



Some person type records, like tenant records, may not have contacts. To email a tenant, set up YSR to send an email to the email ID of the person record (the tenant record in this case) rather than the person record's contacts.

## To set up attachment and email

- Select **Admin > YSR Correspondence > Setup Report**.



You can also access the **YSR Correspondence Setup** screen from the **YSR Admin** side menu: **Setup > Setup Report**. For information about this menu, see “Adding the YSR Admin Menu” on page 162.

- Click **Submit**. The **YSR Report Setup** screen appears.

- Click the **Attachments & Email** tab.

The screenshot shows the 'Report Setup' screen with the 'Attachments & Email' tab selected. The 'Default Options' section contains fields for 'Attach Reports?' (dropdown), 'Email Reports?' (dropdown), 'Merge Reports with No Email-ID' (checkbox), 'Email Output To (Specify Email-ID)' (text input), and 'Require Preview' (checkbox). The 'Report Output Path' section has a 'Output Path' field. At the bottom are file attachment icons.



The fields in the **Default Options** section are for defining runtime defaults, not for setting up email and attachment. For example, if you select **Yes** in the **Attach Reports** field, you must still complete the rest of this procedure before you can attach reports.

- Click the **New Record** button. The **Attachment & Email** screen appears.

The screenshot shows the 'Attachment & Email' screen for a report titled 'Commercial Charges (CommCh)'. It includes sections for 'Object Type Setup' (Object Type: Property, ID Column: propertyid), 'Attachment Setup' (Attach checked, Attachment Type dropdown, Email checked, Email Template: Email To Property Manager, Email Type: Contact, Email To (Role): Property Manager, Include All Contacts checked), and 'Additional Recipients' (a table with columns: Object Type, ID Column, Email Type, Role, To/Cc/Bcc, Include All Contacts). The table rows show settings for 'Owner' and other roles.

Object Type	ID Column	Email Type	Role	To/Cc/Bcc	Include All Contacts
Owner	ownerid	Owner		Cc	<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>

- Complete the fields. For more information, see “Attachment & Email Screen Reference” on page 128.
- Click **Save**.

- 7 Click **Close**. The configuration record appears in the grid on the **Attachment & Email** tab.

- 8 Complete the fields on the **Attachment & Email** tab. For more information, see “Attachment & Email Tab Screen Reference” on page 126.

- 9 Click **Save**. The report is ready for use.

### Attachment & Email Tab Screen Reference

#### Attach Reports?

Determines the state of the **Attach Reports** check box on the report generation screen.

**Blank** The user can choose whether to attach reports to Voyager objects at runtime.

**Yes** Voyager automatically attaches reports to Voyager objects when the user clicks **Generate**. The user cannot clear the **Attach Reports** check box.

**No** Voyager prevents the user from attaching reports to Voyager objects.

**TIP** To email reports to Voyager contacts, you must also attach them to Voyager objects. Therefore, if you select **Yes** in the **Email Reports** field, select **Yes** in this field as well.

<b>Email Reports?</b>	Determines the state of the <b>Email Reports</b> check box on the report generation screen.  <b>Blank</b> The user can choose whether to email reports at runtime.  <b>Yes</b> Voyager automatically emails reports when the user clicks <b>Generate</b> . The user cannot clear the <b>Email Reports</b> check box.  <b>No</b> Voyager prevents the user from emailing reports.  <b>TIP</b> To email reports to Voyager contacts, you must also attach them to Voyager objects. Therefore, if you select <b>Yes</b> in this field, select <b>Yes</b> in the <b>Attach Reports</b> field as well.
<b>Merge Reports with No Email-ID</b>	If the recipients of one or more reports are missing email addresses, Voyager merges the reports with missing emails into one document.  <b>TIP</b> This option makes it easier to deal with large batches of correspondence in which several reports may not have working email addresses. Voyager merges the problem reports together so you can open one document, print in one batch, and mail.
<b>Show on Portal?</b>	Determines the state of the <b>Show On Portal</b> check box on the report generation screen.  <b>Blank</b> The user can choose whether to check the <b>Weblinks</b> box on the <b>Object Attachments</b> screen, thereby publishing the attachment to the Portal site.  <b>Yes</b> Voyager automatically checks the <b>Weblinks</b> box on the <b>Object Attachments</b> screen when the user clicks <b>Generate</b> . The user cannot clear the <b>Show On Portal</b> check box.  <b>No</b> Voyager prevents the user from checking the <b>Weblinks</b> box on the <b>Object Attachments</b> screen, thereby not publishing the document on the Portal site.  <b>TIP</b> To check the <b>Weblinks</b> box, and thus show attachments on the Portal site, you must also attach them to Voyager objects. Therefore, if you select <b>Yes</b> in this field, select <b>Yes</b> in the <b>Attach Reports</b> field.  <b>NOTE</b> To select the <b>Show On Portal</b> check box, users must have access to the <b>Correspondence: YSR - Show on Portal</b> permission.
<b>Email Output To (Specify Email-ID)</b>	(Optional). The destination of the report. This field is useful when you know the recipient's email address and you want to enter the email address manually.  <b>TIP</b> This option is useful, for example, if you want to skip the top-level select and key column setup and test a report by sending it directly to a report recipient. Alternatively, use it to add an additional address as a recipient for all rendered reports.
<b>Report Output Path</b>	(Optional). If you want to save rendered reports to a location other than the Reports path, enter the location here. For example:  \\SharedLocation\\Reports\\YSRExports  <b>NOTE</b> Changing the report output path does not change the attachment path.
<b>Require Preview</b>	The <b>Preview</b> button appears on the report generation screen. The first time the user generates the report, the <b>Generate</b> button is deactivated until the user clicks <b>Preview</b> .

**Publish To SharePoint  
(Publish Source)** If your administrator has loaded the optional package to publish to SharePoint, this section appears after you add an attachment and email record. The package is: Opt\_ss\_Intl\_CorrespondenceEnablePublishToSharepoint.pkg

To publish to SharePoint, enter the source name of the SharePoint location to which Voyager uploads the report.

**NOTE** Complete this field carefully. The text entered here must match exactly the name of the path where you want to upload the report. Contact your SharePoint administrator to identify the source name for your report.

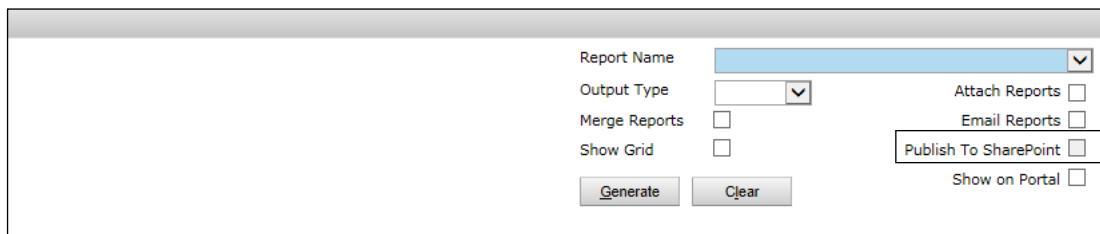
## Attachment & Email Screen Reference

<b>Inactive</b>	Deactivates the email and attachment record.
<b>Object Type</b>	The object type of your report.
<b>ID Column</b>	The alias of the primary key (the hmy) of your report. Use the same alias as used in the top-level select statement.  <b>TIP</b> The top-level select statement of your report must retrieve the object hmy, and must use the hmy as the key column. Enter this key column alias in the <b>ID Column</b> field.
<b>Attach</b>	Determines the state of the <b>Attach Reports</b> check box on the report generation screen.  <b>Blank</b> The user can choose whether to attach reports to Voyager objects at runtime.  <b>Yes</b> Voyager automatically attaches reports to Voyager objects when the user clicks <b>Generate</b> . The user cannot clear the <b>Attach Reports</b> check box.  <b>No</b> Voyager prevents the user from attaching reports to Voyager objects.  <b>TIP</b> To email reports to Voyager contacts, you must also attach them to Voyager objects. Therefore, if you select <b>Yes</b> in the <b>Email Reports</b> field, select <b>Yes</b> in this field as well.

<b>Attachment Type</b>	Voyager applies the attachment type selected here to the attachment record. A label for classifying attachments.
<b>Email</b>	<p>Determines the state of the <b>Email Reports</b> check box on the report generation screen.</p> <p><b>Blank</b> The user can choose whether to email reports at runtime.</p> <p><b>Yes</b> Voyager automatically emails reports when the user clicks <b>Generate</b>. The user cannot clear the <b>Email Reports</b> check box.</p> <p><b>No</b> Voyager prevents the user from emailing reports.</p> <p><b>TIP</b> To email reports to Voyager contacts, you must also attach them to Voyager objects. Therefore, if you select <b>Yes</b> in this field, select <b>Yes</b> in the <b>Attach Reports</b> field as well.</p>
<b>Email Template</b>	<p>The email template Voyager uses.</p> <p><b>TIP</b> An email template is required in order to email reports to Voyager contacts.</p> <p>For more information about setting up email, see “E-Mail Templates” in the <i>Voyager Workflows and Notifications User’s Guide</i>.</p>
<b>Email Type</b>	<p>Determines whether Voyager sends email directly to the email ID associated with a person record or to the contacts associated with a role.</p> <p><b>TIP</b> The options available in this field depend on your selection in the <b>Object Type</b> field. All object types have contacts; some object types also have person records. If you select <b>Contact</b>, the <b>Email To (Role)</b> drop-down list becomes active.</p> <p><b>TIP</b> To send an email to a tenant, select <b>Tenant</b> in the <b>Object Type</b> field and the <b>Email Type</b> field.</p>
<b>Email To (Role)</b>	<p>This field becomes active if you select <b>Contact</b> from the <b>Email Type</b> drop-down list.</p> <p>The role of the contact to whom Voyager sends the report.</p> <p>For more information about contacts, see the <i>Voyager Workflows and Notifications User’s Guide</i>.</p>
<b>Include All Contacts</b>	<p>This field becomes active if you select <b>Contact</b> from the <b>Email Type</b> drop-down list.</p> <p>Voyager emails reports to all contacts in the roles identified in the <b>Email To (Role)</b> field. For example, if you have two contacts in the Property Manager role and you want to email them both, select this check box.</p>

## Publishing to SharePoint

If you enable the option to publish reports to SharePoint Online, a new check box appears on the report generation screen.



To activate SharePoint for YSR reports:

- 1 Load the following packages:

- Opt\_ss\_Intl\_CorrespondenceEnablePublishToSharepoint.pkg
  - Opt\_ss\_YSR\_SPOnline\_Option\_To\_Publish.pkg
- 2 Enter the source name of your SharePoint location on the **Attachment & Email** tab on the **YSR Report Setup** screen. For more information, see “Attachment & Email Tab Screen Reference” on page 126.

Document Management for Sharepoint users can publish reports directly to Sharepoint Online from the report generation screen by selecting the **Attach Reports** check box in combination with the **Publish to SharePoint** check box.

OwnerId	OwnerCode	Owner Name	Address	City	EmailID	PropertyID	Report	Email
72.00	glegal	Mike Delia	Gorden house	Sampson	glegal@yardi.com	21.00	<a href="#">View Report</a>	<a href="#">View Email</a>
72.00	glegal	Mike Delia	Gorden house	Sampson	glegal@yardi.com	22.00	<a href="#">View Report</a>	<a href="#">View Email</a>
72.00	glegal	Mike Delia	Gorden house	Sampson	glegal@yardi.com	170.00	<a href="#">View Report</a>	<a href="#">View Email</a>



You must set up the report to create attachments, and the user must select the **Attach Reports** check box when generating the report.

For more information about setting up attachments, see “To set up attachment and email” on page 125.

Reports appear in SharePoint Online:

Name	Title	Owner Code	Owner First Name
ClientStatement_glegal_0_5_8_2...	ClientStatement glegal 0 5 8 2018 16 51 35		
ClientStatement_glegal_0_5_8_2...	ClientStatement glegal 0 5 8 2018 16 51 24		
ClientStatement_glegal_0.xlsx	ClientStatement glegal		

## Publishing to Portal

Portal users can publish documents to Portal directly from the YSR report generation screen.

investorid	Investor Code	Investor Name	Investment Code	Investment Code1	Report
136.00	cdinv	Canada Investor	nifund	Netherlands Fund	<a href="#">View Report</a>
137.00	nlinv	Netherlands Investor	nifund	Netherlands Fund	<a href="#">View Report</a>



To select the **Show On Portal** check box, users must have access to the **Correspondence: YSR - Show on Portal** permission.



You must set up the report to create attachments, and the user must select **both** the **Attach Reports** check box and the **Show on Portal** check box when generating the report.

For more information about setting up attachments, see “To set up attachment and email” on page 125.

You can verify that the document appears on Portal by reviewing the **Attachments** screen (**Functions > Attachments**) of the related Voyager object. Voyager selects the **Show on Portal** screen to indicate that the document is available on Portal.

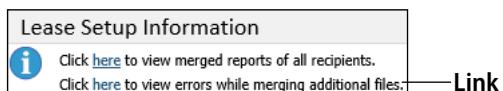
Type	Description	Date	Attachment	Secure	Show on Portal	Detach
	scapcall_136_0	06/09/2016	scapcall_136_0.pdf	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	scapcall_136_0	06/09/2016	scapcall_136_0_1.pdf	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

## Holding Emails

To prevent YSR from sending emails upon report generation, select the **Hold Emails** check box. If you want to send the emails, clear the check box and generate the reports again.

## Static PDF Attachment Error Logs

To view error logs related to static PDF attachments when previewing YSR reports, click the **here** link that appears on the top part of the screen.



The graphic below shows an example error log.

```
Report : [Word0002] Lease Setup Information
User :

-----
LeaseCode='t0000045' [2020-11-02 03:53:50 PM]
Error: Exception occurred during the processing file: 5
[1]\3\voyager7s\Reports\temp\Word0002_t0000045_LeasInfo_t0000045_4.pdf
[2]\3\voyager7s\Reports\attachment\CommLease\10-2020\Word2_t0000045_4_2.pdf
[3]\3\voyager7s\Reports\attachment\CommLease\10-2020\Word2_t0000045_4_3.pdf
```

## Monitoring Reports

If you want to monitor user activity, you can load a package to start tracking report generation in the YSRReportHistory table (load Opt\_ss\_YSR\_EnableReportHistory.pkg). The YSRReportHistory table provides a time-stamped record each time a user generates a report, as illustrated in the following graphic:

	hMy	sCode	hUser	dtStart	sParms	dtLastUpdate	sMessage	bComplete
1	1	ASrTest	0	2018-04-12 15:49:29.303	&property=31	2018-04-12 15:49:33.910		-1
2	2	ASrTest	0	2018-04-12 15:49:47.203	&property=31	2018-04-12 15:49:47.233		-1
3	3	RentRoll	0	2018-04-12 15:51:12.353	&PropertyList=2&FromD...	2018-04-12 15:51:13.673		-1
4	4	RentRoll	0	2018-04-12 15:52:46.983	&PropertyList=2&FromD...	2018-04-12 15:52:47.060		-1
5	5	RentRoll	0	2018-04-12 15:52:58.403	&PropertyList=2, 108&F...	2018-04-12 15:52:58.403	NULL	0
6	6	ASrTest	0	2018-04-12 15:54:14.053	&property=31	2018-04-12 15:54:14.073		-1
7	7	RentRoll	0	2018-04-12 15:54:59.773	&PropertyList=2, 108&F...	2018-04-12 15:54:59.773	NULL	0
8	8	ASrTest	0	2018-04-12 16:24:57.663	&property=31	2018-04-12 16:24:57.687		-1
9	9	RentRoll	0	2018-04-12 16:30:12.563	&PropertyList=2, 108&F...	2018-04-12 16:30:12.667		-1
10	10	ASrTest	0	2018-04-12 16:52:45.317	&property=31	2018-04-12 16:52:45.340		-1
11	11	ASrTest	0	2018-04-13 09:00:53.947	&property=2	2018-04-13 09:00:56.153		-1
12	12	ASrTest	0	2018-04-13 09:30:15.510	&property=2	2018-04-13 09:30:15.533		-1
13	13	AP	0	2018-04-13 09:32:19.780	&Owner=1629 &Property...	2018-04-13 09:32:41.017	The report has ...	-2

The bComplete column returns -1 for successful report generation, 0 for a report that fails without generating an error message, and -2 for a report that fails with an error message. Error messages, if available, are stored in the sMessage column.

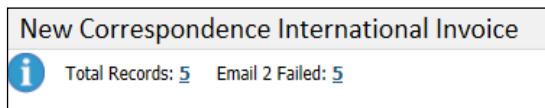
The sParms column tracks all filter criteria associated with the report, but it does not show publication parameters (things like output type, show on portal, show grid, and others). For publication-related parameters, review the iFilter table.

If you are interested in tracking YSR reports that are generated with Report Scheduler, there is a separate monitor for that purpose. For more information, see "Report Scheduler" on page 151

## Reviewing YSR Report Generation History

Voyager stores the total counts of generated and emailed YSR reports in the YSRReportHistory table. You can load a package if you want to show this information on the report generation screen (load Opt\_ss\_YSR\_EnableReportHistory.pkg).

**Total Records** and **Email Failed** counts appear at the top of the report generation screen.



You can click the hyperlinks to open the **YSR - Generate Report History** screen, which provides summarized and detailed information about attached and emailed reports.



You can also access the **YSR - Generate Report History** screen from the **YSR Admin** side menu: **Reports > Generate Report History**.

The **Summary** tab displays whether the report was successfully attached or emailed.

YSR - Generate Report History						
Summary		Attachment-Email Details				
Report Name	Status	Attachment	Attachment 5	Email	Email 2	Email 3
New Correspondence International Invoice (CInvoice)	Successful	3	3	2	1	
	Failed			1	3	2

If you want to view more details, including Voyager objects attached by record code, you can click the **Attachment-Email Details** tab.

YSR - Generate Report History										
Summary		Attachment-Email Details								
Record Code	Attachment/Email	Object Type	Attachment		Email		Date Sent	Subject	To	Sender
			Attachment Type	Date Attached	Email					
201000000001	Attachment/Email 1	Owner	CInvoice_201000000001_4.xlsx	21.09.2018 05:40 PM	Email address for recipient is not set up.					
	Attachment/Email 2	Customer			Email address for recipient is not set up.					
	Attachment/Email 3	Commercial Lease			Email address for recipient is not set up.					
	Attachment/Email 5	Owner	CInvoice_201000000001_4.xlsx	21.09.2018 05:40 PM						
	Attachment/Email 6	Owner	CInvoice_201000000004_4.xlsx	21.09.2018 05:40 PM						
201000000004	Attachment/Email 1	Owner	CInvoice_201000000004_4.xlsx	21.09.2018 05:40 PM						
	Attachment/Email 2	Customer			Email address for recipient is not set up.					
	Attachment/Email 3	Commercial Lease			Email address for recipient is not set up.					
	Attachment/Email 5	Owner	CInvoice_201000000004_4.xlsx	21.09.2018 05:40 PM						
	Attachment/Email 6	Owner	CInvoice_201000000048_4.xlsx	21.09.2018 05:40 PM						
201000000048	Attachment/Email 1	Owner	CInvoice_201000000048_4.xlsx	21.09.2018 05:40 PM						
	Attachment/Email 2	Customer			Email address for recipient is not set up.					
	Attachment/Email 3	Commercial Lease			Email address for recipient is not set up.					
	Attachment/Email 5	Owner	CInvoice_201000000048_4.xlsx	21.09.2018 05:40 PM						

# Output Options

## In this section:

Displaying Graphical Elements on Screen.....	134
Generating Ad Hoc Reports in Report Scheduler.....	137
Generating QR Codes Dynamically at Runtime.....	139

This section describes how to configure output options. Options include publishing graphical elements to screen (Excel-based reports only), sending reports to Report Scheduler, and generating QR codes dynamically at runtime.

## Displaying Graphical Elements on Screen

When you design your report templates in Excel, you can take advantage of several features of Excel formatting when publishing to screen. You can turn on these features by selecting the **Enable New Screen Output** check box on the **Report Options** tab on the **YSR Report Setup** screen.

The screenshot shows the YSR Report Setup interface. At the top, there's a toolbar with 'File', 'Edit', 'View', 'Format', 'Report', 'Help', and a 'Search' field. Below the toolbar, the main area has a title bar 'Merged Report Setup'. Under this, there are several input fields and buttons:

- Code:** CommCh
- Description:** Comm Charges
- Select Statement:** SELECT p.scode property, p.sAddr1 PropName FROM Property p where 1=1
- OR Script File:** (empty)
- Key Column:** property
- Inactive:**
- VoyagerPlus Report:**
- Buttons:** Save, New, Define Filters, Map Filters, Dump SQL, Dump Oracle, Delete Setup.

To the right of these fields is a 'Notes' text area with a scroll bar.

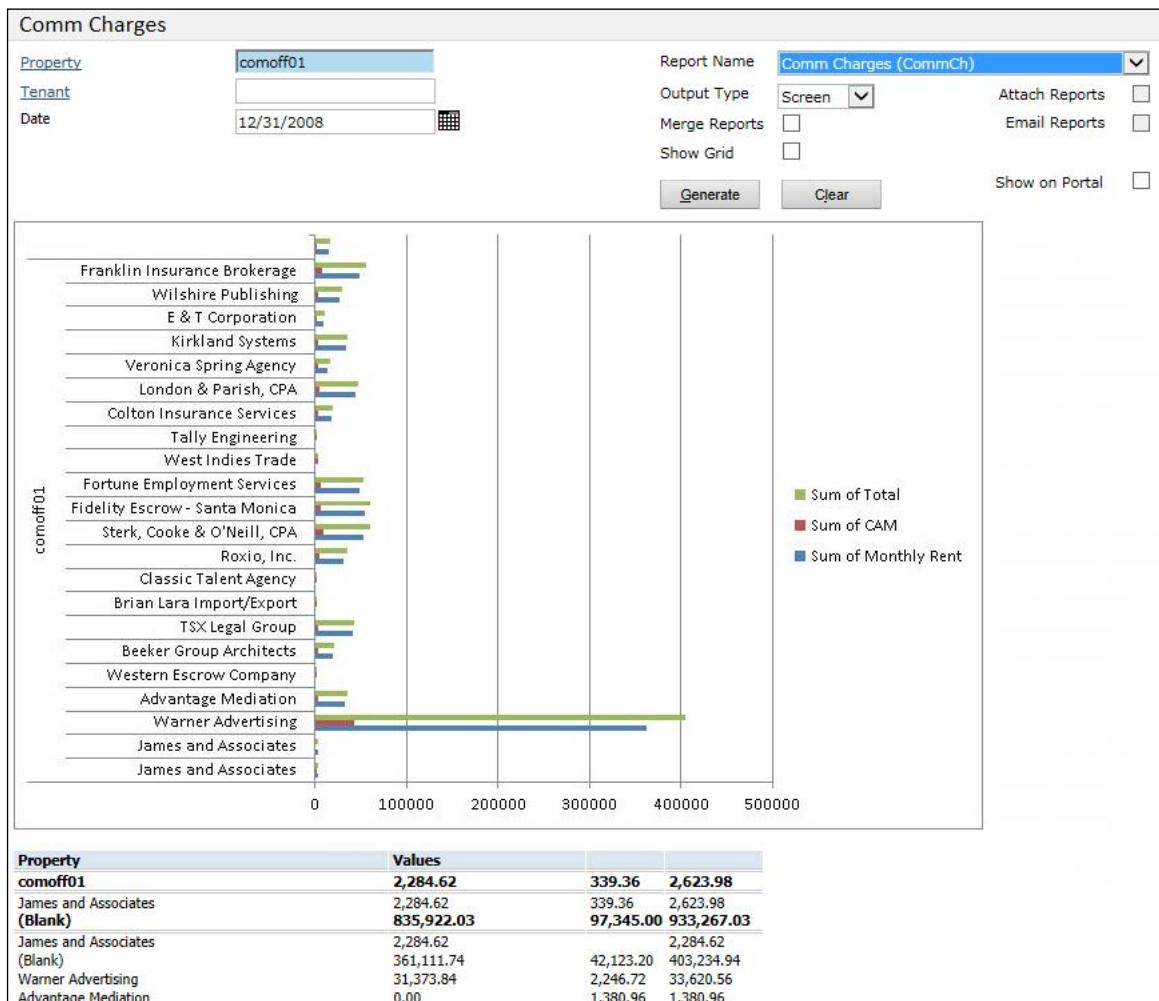
Below the main setup area is a navigation bar with tabs: Report Setup, Attachment & Email, Additional Roles, and Output Options. The 'Output Options' tab is currently selected.

Under the 'Output Options' tab, there's a section titled 'Screen Output Options' with the following settings:

Setting	Status
Enable New Screen Output	<input checked="" type="checkbox"/>
Enable Tabbed Screen Output	<input checked="" type="checkbox"/>
Use Freeze-Pane Setting for Report Title/Header	<input checked="" type="checkbox"/>

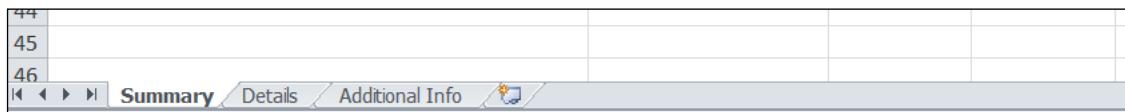
## Pivot tables and graphs on screen

You can display charts and graphs based on pivot table data on screen. For example, the following graphic shows a report that features a pivot table and bar chart.



### Tabbed reports on screen

If your Excel report template has multiple tabs, you can display the tabs on screen. For example, suppose your report template has these tabs:



When published to screen, tabbed data appears:

**Comm Charges**

<b>Property</b> comoff01  <b>Tenant</b>  <b>Date</b> 12/31/2008	<b>Report Name</b> Comm Charges (CommCh) <b>Output Type</b> Screen <input type="checkbox"/> Merge Reports <input type="checkbox"/> Show Grid	<input type="checkbox"/> Attach Reports <input type="checkbox"/> Email Reports <input type="checkbox"/> Show on Portal
<input type="button" value="Generate"/> <input type="button" value="Clear"/>		

---

[Summary](#) | [Details](#) | [Additional Info](#)

Property	Tenant	Date	Monthly Rent	CAM	Total
comoff01	James and Associates	12/31/2008	2,284.62	339.36	2,623.98
	Warner Advertising	12/31/2008	15,686.92	2,246.72	17,933.64
	Advantage Mediation	12/31/2008	0.00	1,380.90	1,380.96
	Western Escrow Company	12/31/2008	17,941.00	2,469.80	20,410.80
	Beeker Group Architects	12/31/2008	9,861.48	1,391.04	11,252.52
	TSX Legal Group	12/31/2008	0.00	854.56	854.56
	Brian Lara Import/Export	12/31/2008	0.00	863.52	863.52
	Beeker Group Architects	12/31/2008	9,861.48	1,391.04	11,252.52
	Classic Talent Agency	12/31/2008	30,546.81	4,371.36	34,918.17

## Bold Header

You can apply bold formatting and increase the font size of the first row of an Excel spreadsheet by selecting the **Use Freeze-Pane Setting for Report Title/Header** check box on the **Output Options** tab.

The screenshot shows the 'Comm Charges' report configuration screen. At the top, there are input fields for 'Property' (comoff01), 'Tenant' (empty), and 'Date' (12/31/2008). To the right are report settings: 'Report Name' (Comm Charges (CommCh)), 'Output Type' (Screen), 'Merge Reports' (unchecked), 'Show Grid' (unchecked), and several checkboxes for 'Attach Reports', 'Email Reports', and 'Show on Portal'. Below these are two tabs: 'ABC Corporation Header' (selected) and 'ABC Corporation Data'. The 'ABC Corporation Header' tab displays a bolded header row with columns: Property, Tenant, Date, Monthly Rent CAM, and Total. The 'ABC Corporation Data' tab displays a grid of data rows with the same columns.

Property	Tenant	Date	Monthly Rent CAM	Total
comoff01			6,920.40	6,920.40
	Franklin Insurance Brokerage	12/31/2008	6,920.40	6,920.40
	E & T Corporation	12/31/2008	13,186.88	13,186.88
	Brian Lara Import/Export	12/31/2008	5,628.30	5,628.30
	Colton Insurance Services	12/31/2008	4,873.28	4,873.28
		12/31/2008	16,334.80	16,334.80
	West Indies Trade	12/31/2008	4,957.68	4,957.68
	Sterk, Cooke & O'Neill, CPA	12/31/2008	13,814.08	13,814.08
		12/31/2008	4,116.00	4,116.00
	Advantage Mediation	12/31/2008	8,877.60	8,877.60
	Warner Advertising	12/31/2008	15,686.92	15,686.92
	James and Associates	12/31/2008	2,284.62	2,284.62
	Beeker Group Architects	12/31/2008	5,041.90	5,041.90
	TSX Legal Group	12/31/2008	5,417.30	5,417.30
	Beeker Group Architects	12/31/2008	4,819.58	4,819.58
	Veronica Spring Agency	12/31/2008	11,349.80	11,349.80
		12/31/2008	4,958.80	4,958.80
	Fidelity Escrow - Santa Monica	12/31/2008	5,857.80	5,857.80
		12/31/2008	10,101.00	10,101.00
	Mortgage.com	12/31/2008	0.00	0.00

## Generating Ad Hoc Reports in Report Scheduler

This section describes how to give users the option to send YSR reports to Report Scheduler on an ad hoc basis from the **Report Generation** screen.

When you turn on this option, new options appear in the **Output Type** field (**Send PDF**, **Send Excel**, and **Send Word**). The **Email Address** field also appears, where the user can enter a report destination.

The screenshot shows the 'Unit Directory' report configuration screen. It has input fields for 'Property' (comoff01) and 'Building' (empty). To the right are report settings: 'Report Name' (Unit Directory (UnitDir)), 'Output Type' (Send PDF), 'Merge Reports' (checked), 'Show Grid' (unchecked), and checkboxes for 'Attach Reports', 'Email Reports', 'Publish To SharePoint', and 'Show on Portal'. Below these are two tabs: 'Unit Directory' (selected) and 'Unit Directory Data'. The 'Unit Directory' tab displays a grid of data rows with columns: Building, Address, City, State, Zip, and Phone. The 'Unit Directory Data' tab displays a grid of data rows with columns: Building, Address, City, State, Zip, and Phone. An 'Email Address' field at the bottom contains the value 'UsersBackupEmail@corporation.com'.

Building	Address	City	State	Zip	Phone

When the user clicks **Generate**, the YSR report runs immediately in Report Scheduler.

Report Scheduler sends the report to all recipients indicated in the report design as well as to any address the user enters in the **Email Address** field at runtime.

## To generate ad hoc reports in Report Scheduler

- 1 Make sure that rs\_YSR\_CommonConductorInterface.txt exists in your Reports path.
- 2 Open the YSR report.
- 3 Click the **Output Options** tab.

The screenshot shows the 'Output Options' tab selected in the top navigation bar. The interface is divided into several sections with blue headers:

- Screen Output Options**: Contains three checkboxes: 'Enable New Screen Output', 'Enable Tabbed Screen Output', and 'Use Freeze-Pane Setting for Report Title/Header'. All are unchecked.
- Scheduler Report Setup**: Contains three checkboxes: 'Regular Report', 'Long Running Report' (which is checked), and a 'Long Running Script File Name' input field containing the value 'LongRunningReport'.
- QR Code Setup**: Contains a 'QR Code Section Columns' input field.
- Report Generate Help**: Contains a 'Help Topic' input field.

- 4 If you are setting up a regular report (one that does not cause timeout problems when generated using normal procedures), select the **Regular Report** check box.
- 5 If you are setting up a resource-intensive report (one that may cause timeout problems when generated using normal procedures), select the **Long Running Report** check box.
- 6 Click **Save**. If you selected the **Long Running Report** check box, Voyager generates a .txt file and displays the filename in the **Long Running Script File Name** field.

The screenshot shows the 'Output Options' tab selected in the top navigation bar. The interface is divided into several sections with blue headers:

- Screen Output Options**: Contains three checkboxes: 'Enable New Screen Output', 'Enable Tabbed Screen Output', and 'Use Freeze-Pane Setting for Report Title/Header'. All are unchecked.
- Scheduler Report Setup**: Contains three checkboxes: 'Regular Report', 'Long Running Report' (which is checked), and a 'Long Running Script File Name' input field containing the value 'rs\_YSR\_CommonConductorInterface\_1098\_YSR.txt'.
- QR Code Setup**: Contains a 'QR Code Section Columns' input field.
- Report Generate Help**: Contains a 'Help Topic' input field.

When users select the report on the **Report Generation** screen, new publishing options appear (**Send Excel [Word]** and **Send PDF**).

- 7 Generate the report.

## Generating QR Codes Dynamically at Runtime

If your organization uses Quick Response (QR) codes, for example to print two-dimensional bar codes on invoices, you can configure YSR reports to generate QR codes dynamically at runtime.

### To generate QR codes dynamically

- 1 Create a script that generates the raw text that you want to encode. The string should contain identifying data like property hmys or scodes. For example, you might encode a raw string like the following:

<https://y.com?PropertyID=2&PropertyCode=comoff01>

An example script appears below (section name and relevant field alias highlighted in bold):

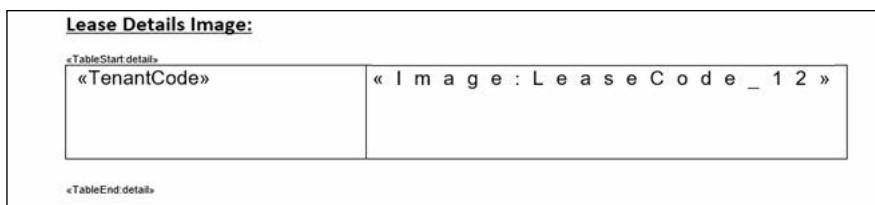
```
//SELECT detail
Select p.hmy PropertyId
,p.scode PropertyCode
,t.scode TenantCode
,'https://y.com?PropertyID=' + convert(varchar,p.HMY) +
'&PropertyCode=' + p.scode LeaseCode_12
from property p
inner join tenant t on t.hproperty = p.hmy
//END SELECT
```

- 2 Add to your report template an image smart marker (Excel) or merge field (Word).

Excel:

A	B	C
1 &=header.PropLogo(picture:FitToCell)		
2		
3		
4		
5		
6		
7 &=detail.LeaseCode_12(picture)		
8		
9		

Word:



- 3 On the **YSR Report Setup** screen, complete the **QR Code Setup** section of **Output Options** tab.

Report Setup	Attachment & Email	Output Options	Digital Signature
<p><b>Screen Output Options</b></p> <p>Enable New Screen Output <input type="checkbox"/></p> <p>Enable Tabbed Screen Output <input type="checkbox"/></p> <p>Use Freeze-Pane Setting for Report Title/Header <input type="checkbox"/></p> <p><b>Scheduler Report Setup</b></p> <p>Regular Report <input checked="" type="checkbox"/></p> <p>Long Running Report <input type="checkbox"/></p> <p>Long Running Script File Name <input type="text"/></p> <p><b>QR Code Setup</b></p> <p>QR Code Section Columns <input type="text" value="detail.LeaseCode_12"/></p> <p><b>Report Generate Help</b></p> <p>Help Topic <input type="text"/></p>			

**QR Code Section Columns** Enter the SectionCode.FieldAlias of the data column containing the raw data you want to encode.

**TIP** If you have multiple QR codes in your report, enter the codes separated by a caret (^). For example:  
detail.LeaseCode\_12^header.PropertyQRCode

- 4 Generate the report. QR codes appear in the rendered report.



- 5 Decode the QR codes. The decoder reveals the raw text generated dynamically by your script at runtime.

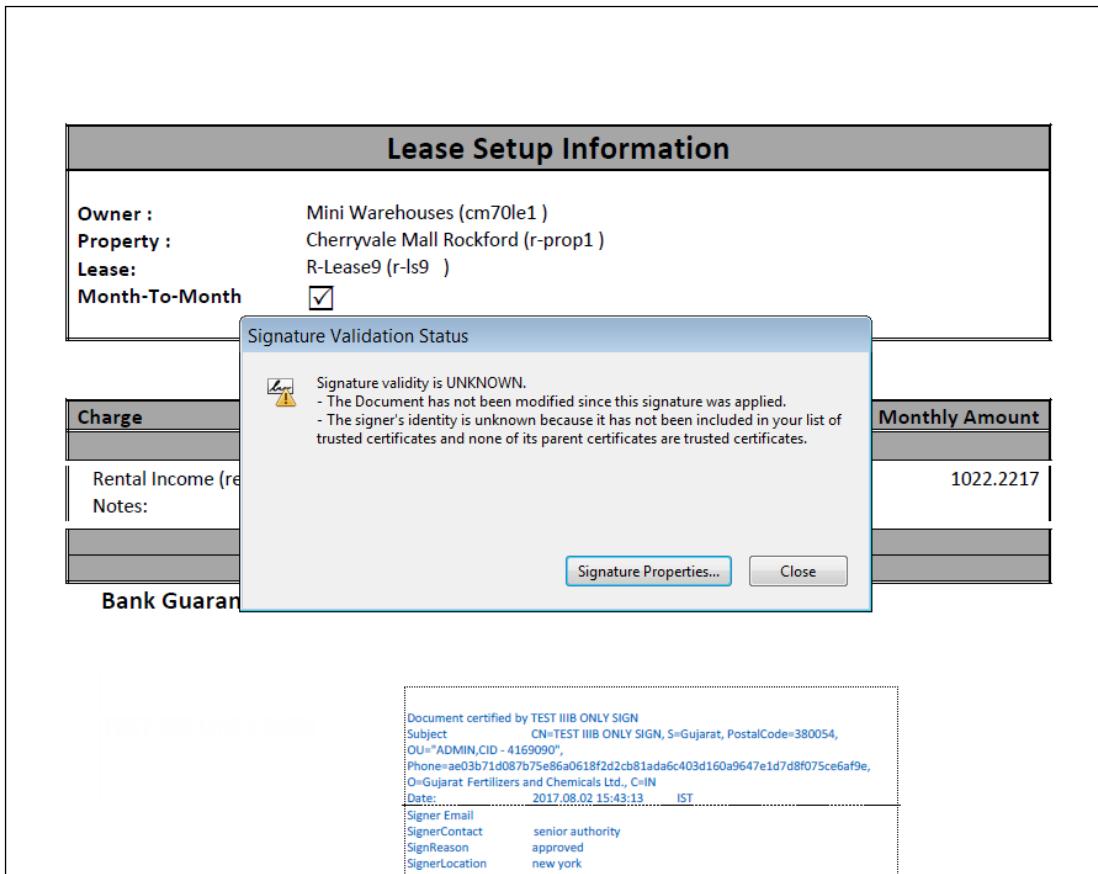
 Decode Succeeded	
Raw text	<a href="https://y.com?PropertyId=79&amp;PropertyCode=anprop01">https://y.com?PropertyId=79&amp;PropertyCode=anprop01</a>
Raw bytes	43 16 87 47 47 07 33 a2 f2 f7 92 e6 36 f6 d3 f5 07 26 f7 06 57 27 47 94 96 43 d3 73 92 65 07 26 f7 06 57 27 47 94 36 f6 46 53 d6 16 e7 07 26 f7 03 03 10 ec 11 ec 11
Barcode format	QR_CODE
Parsed Result Type	URI
Parsed Result	<a href="https://y.com?PropertyId=79&amp;PropertyCode=anprop01">https://y.com?PropertyId=79&amp;PropertyCode=anprop01</a>

## Digital Signatures

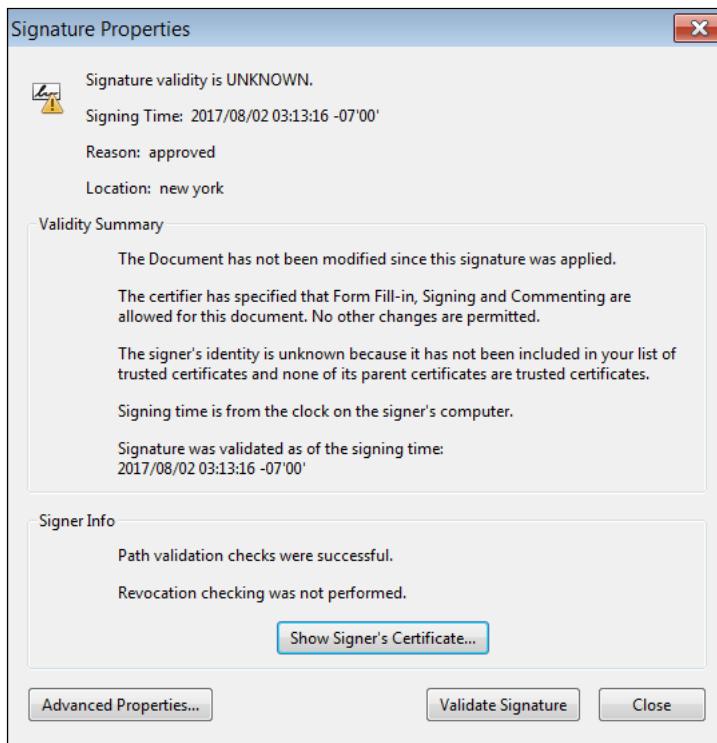
You can embed digital signature technology in Word-based YSR reports (owner- and property-based reports only). Digital signatures use Personal Information Certificates (.pfx files) to provide the recipients of a download a way to verify the source and integrity of the download. Set up digital signatures if your YSR report recipients require a way to verify the authenticity of your reports.

When you set up a YSR report with a digital signature file, the rendered report (Word or PDF) features a click-able text box containing text and tokens that you specify. On click, a dialog box appears where the report recipient can review the document's certificate of authenticity and validate it (add it to a list of trusted certificates).

The following graphic shows an example YSR report with embedded digital signature technology. On click, a dialog box appears:



The report recipient can click the **Signature Properties** button to review information about your security certificate and its issuance chain.



If the user adds your security certificate to their list of trusted certificates, then all documents with your digital signature are identified for the user as having a valid signature. When a user clicks the digital signature area, the signature is marked as valid.



This document assumes that your organization has a .pfx file and accompanying certificate (one or more .cer, .cert, or .crt files). For information about creating a PFX certificate, see Microsoft's online help.

## Options for digitally signing YSR reports

Once you complete digital signature setup tasks for all the objects (properties or owners) that require signatures, all YSR reports are signed when a user generates the report.

You can, however, set up the report so that Voyager users are required to locate the appropriate PFX file and enter a password before they can generate the report. The following graphic shows the new fields that appear to users if they are required to sign the report before generating it:

The screenshot shows a report setup dialog box titled 'Comm Charges Demo'. On the left, there are input fields for 'Property' (containing 'comoff01'), 'Tenant' (empty), and 'Date' (containing '12/31/2008'). On the right, there are several configuration options: 'Report Name' (set to 'Comm Charges Demo (CommCh)'), 'Output Type' (set to 'Screen'), 'Merge Reports' (unchecked), 'Show Grid' (unchecked), 'Attach Reports' (checked), 'Email Reports' (checked), 'Publish To SharePoint' (unchecked), and 'Show on Portal' (unchecked). Below these are two input fields: 'Signature File' (empty) with a 'Browse...' button, and 'Password' (empty). At the bottom are three buttons: 'Generate', 'Clear', and 'Help'.

The Voyager user must enter the file location and password associated with the reporting object, as configured on the **Digital Signature Setup** screen. For more information, see “Setting up Digital Signatures” on page 144.

To require Voyager users to digitally sign at runtime, select the option to **Enable to select signature file on generate** option on the report setup screens. For details, see “Adding Digital Signatures to YSR Reports” on page 146.



Do not require a Voyager user to digitally sign at runtime if you plan to process the report through Task Runner or Report Scheduler.

## Required setup steps

To embed digital signature technology in a Word-based YSR report, complete the following steps:

- Set up digital signatures (associate a PFX certificate and password with one or more Voyager objects).
- Add digital signatures to your YSR report (edit the **YSR Report Setup** screen).
- Edit your report template (add digital signature tokens to your Word report template).

## Setting up Digital Signatures

When you set up digital signatures for YSR reports, you create an association between the location of your PFX certificate, a password, and one or more Voyager objects (owners or properties only).

These associations enable Voyager to identify the correct PFX certificate for each object, as filtered by the user at runtime.

To set up digital signatures, you must access the **Digital Signature Setup** page at this location:

[./pages/CustomCorrespDigitalSignatureSetup.aspx](#)



You can also access the **Digital Signature Setup** screen from the **YSR Admin** side menu: **Setup > Digital Signature**. For information about this menu, see “Adding the YSR Admin Menu” on page 162.

Set up a custom menu path to access this page quickly from your menu set.

Object Type	Record Code	Delete
Owner	glegal	<input type="button" value="Delete"/>
Owner		<input type="button" value="Delete"/>

#### Signature File

Click **Browse** to locate the PFX certificate you want to associate with a password and one or more Voyager objects.

Depending on how you set up your YSR report, a Voyager user may be required to identify the correct PFX document for their reporting object before they can generate the report.

#### Password

Add a password.

Depending on how you set up your YSR report, a Voyager user may be required to enter the correct password for their reporting object before they can generate the report.

#### Object Type

Properties and owners are supported.

#### Record Code

Identify the code of the owners or properties you want to associate with the PFX certificate.

## Adding Digital Signatures to YSR Reports

To embed digital signature technology in YSR report, complete the **Digital Signature** tab on the **YSR Report Setup** screen. This tab contains information that Voyager uses at runtime to render the report and the clickable signature area.

The screenshot shows the 'YSR Report Setup' window. At the top, there are fields for 'Code' (Word0002), 'Description' (Lease Setup Information), 'Top-Level Select Statement' (Select o.hmyperson ownerid, o.uCode OwnerCode), 'Key Column' (LeaseCode), and 'Inactive' (unchecked). Below these are buttons for 'Save', 'New', 'Save Copy', and several action buttons: 'Define Filters', 'Map Top-Level Filter', 'Dump SQL', 'Delete Setup', and 'Help'. A 'Notes' text area is also present. The bottom section is titled 'Digital Signature' and includes fields for 'Object Type' (Owner), 'ID Column' (ownerid), 'Enable to select signature file on generate' (checked), 'Location' (New York), 'Reason' (Approved), 'Contact' (Senior Authority), and settings for 'Height (in pixel)' (150), 'Width (in pixel)' (260), and 'Offset (in pixel)' (1).

<b>ID Column</b>	The alias used in the top-level select for the unique identifier of the digital signature object. The identifier must be either owner.hmyperson (for objects) or property.hmy (for properties).  <b>TIP</b> The unique identifier of the digital signature object does not need to be the same as the report's key column, but it does need to be queried in the top-level select. In this example, the top-level select statement retrieves owner.hmyperson using the alias <b>ownerid</b> . The report key column, by contrast, is LeaseCode.
<b>Enable to select signature file on generate</b>	When the user selects this YSR report on the report generation screen, new fields appear where the user must browse to the digital signature (PFX) file and enter a password before generating the report.  <b>TIP</b> Select this option if you want Voyager users to be required to digitally sign the report before generating it. Do not select this option if you want to generate the report in Task Runner or Report Scheduler.
<b>Location</b>	Corresponds to the <<signerlocation>> token. If you use the <<signerlocation>> token in your report template, Voyager replaces the token with the text in this field when you generate the report.
<b>Reason</b>	Corresponds to the <<signerreason>> token. If you use the <<signerreason>> token in your report template, Voyager replaces the token with the text in this field when you generate the report.

Contact	Corresponds to the <<signercontact>> token. If you use the <<signercontact>> token in your report template, Voyager replaces the token with the text in this field when you generate the report.
Height	The height of the click-able digital signature area, placed at the #@@DigitalSignature# token in your report template.
Width	The height of the click-able digital signature area, placed at the #@@DigitalSignature# token in your report template.
Offset	The offset from the table cell boundary of the click-able digital signature area, placed at the #@@DigitalSignature# token in your report template. (You must place the token in a table in your Word template.)

## Adding Digital Signature Tokens to Report Templates

To create a click-able digital signature certificate in a Word report template, add a table to your template and place the following token in it:

#@@DigitalSignature#



You can apply white font color to the token and white background color to the table cell to obscure any background text in the rendered report.

The dimensions of the click-able area in your document are determined by the dimensions you enter on the **Digital Signature** tab on the **YSR Report Setup** screen, as described in the preceding topic.

You can optionally add the following merge fields to your template:

<<SignerSubjectName>>	Replaced at runtime by information from the PFX certificate.
<<SignerName>>	Replaced at runtime by information from the PFX certificate.
<<SignDateTime>>	Replaced at runtime by information from the PFX certificate.
<<SignerEmail>>	Replaced at runtime by information from the PFX certificate.
<<SignerContact>>	Replaced at runtime by the text in the <b>Contact</b> field on the <b>Digital Signature Setup</b> tab on the <b>YSR Report Setup</b> screen.
<<SignerReason>>	Replaced at runtime by the text in the <b>Reason</b> field on the <b>Digital Signature Setup</b> tab on the <b>YSR Report Setup</b> screen.
<<SignerLocation>>	Replaced at runtime by the text in the <b>Location</b> field on the <b>Digital Signature Setup</b> tab on the <b>YSR Report Setup</b> screen.

## Task Runner

Task Runner is a module that you can use to schedule functions to run automatically on a Voyager server. When you add a Task Runner task, you must specify a custom URL for Voyager to use when executing the task. The URL contains parameters and parameter values that provide Voyager with information on how to process a task.

This section provides an example of a custom URL for use in Task Runner. It is not a complete description of Task Runner. For more information on this topic, see the *Voyager Service Manager Administration and Setup Guide* and the *Task Runner APPTASK ClassNames and URL Parameters* guide.

### YSR Task Class Name

YSI.Utils.CustomCorrespondence.dll#YSI.Utils.CustomCorrespondence.AppClasses.yiYSRCorrespondenceTask

### Example URL

ReportCode=ClientST&OutputType=PDF&Merge=Yes&Attach=Yes&Email=Yes&Generate=Yes&Grid=Yes&OwnerId=glegal&AcctID=111-000-00&FromDate=01/2012&ToDate=12/2013

The URL must contain, at minimum, the ReportCode parameter and parameters corresponding to all required filter fields. Refer to filter fields by their field name. You can pass in multiple filter elements by using a caret (&OwnerId=glegal^hlegal).

### Example URL Components

Parameter	Parameter Value	Mandatory	Example
ReportCode	The code of the report you want to generate.	Yes	ClientST
OutputType	PDF, Excel, or Screen. The format in which Voyager generates the report.		PDF
Attach	Yes or No. If &Attach=Yes, Voyager attaches the report to Voyager objects. You must attach reports if you want to email them.		Yes
Email	Yes or No. If &Email=Yes, Voyager emails the report to recipients. If you want to email reports, you must also attach them.		Yes
HoldEmail	Yes or No. If &HoldEmail=Yes, Voyager does not send emails upon report generation.		Yes
Merge	Yes or No. If &Merge=Yes, Voyager merges the reports into a single file.		Yes
Grid	Yes or No. If &Grid=Yes, Voyager shows grid lines for on screen reports.		Yes

Parameter	Parameter Value	Mandatory	Example
OwnerID	In this example, it is necessary to include the OwnerID parameter because this particular report (ClientST) has a mandatory filter for owner codes. "OwnerID" is the field name specified on the <b>Report Filters Setup</b> screen of the YSR report, ClientST.	As specified on the <b>Report Filters Setup</b> screen.	glegal
AcctID	In this example, it is necessary to include the AcctID parameter because this particular report (ClientST) has a mandatory filter for account nodes. "AcctID" is the field name specified on the <b>Report Filters Setup</b> screen of the YSR report, ClientST.	As specified on the <b>Report Filters Setup</b> screen.	111-000-00
FromDate	In this example, it is necessary to include the FromDate parameter because this particular report (ClientST) has a mandatory date range filter. "FromDate" is the field name specified on the <b>Report Filters Setup</b> screen of the YSR report, ClientST.	As specified on the <b>Report Filters Setup</b> screen.	01/2012
ToDate	In this example, it is necessary to include the ToDate parameter because this particular report (ClientST) has a mandatory date range filter. "ToDate" is the field name specified on the <b>Report Filters Setup</b> screen of the YSR report, ClientST.	As specified on the <b>Report Filters Setup</b> screen.	12/2013
ShowOnPortal	Yes or No. If &ShowOnPortal=Yes, Voyager publishes the report to Portal.		Yes/No
EmailOutputTo	Specific email address. YSR emails reports to specified email addresses. To email multiple recipients, separate email addresses using a semicolon.		first.last@email.com; last.first@email.com
PublishtoSharePoint	True or False. If &PublishtoSharePoint=True, Voyager publishes the report to Sharepoint (required Yardi Document Management for SharePoint).		True/False

## Dynamic Date Values

If you turn on the **Correspondence: YSR - Enable Dynamic Date support for Task** optional parameter (Opt\_ss\_YSR\_DynamicDateForTask.pkg), you can use the following dynamic date values with supported TaskRunner URL parameters:

\$today\$	Current date.
\$firstdayofthismonth\$	First day of current month.
\$lastdayofthismonth\$	Last day of current month.
\$boy\$	Beginning of current year.
\$eoys\$	End of current year.
\$boq\$	Beginning of current quarter.
\$eoq\$	End of current quarter.

You can use a value in place of a date or month. For example:

<b>FromDate=\$today\$</b>	FromDate is the current date (MM/DD/YYYY).
<b>PostMonth=\$boq\$</b>	Post Month is the beginning of the current quarter (MM/YYYY).

You can also add or subtract months and days from a value. For example:

<b>FromDate=\$today\$(d,-7)</b>	FromDate is 7 days before the current date (MM/DD/YYYY).
<b>PostMonth=\$boq\$(m,+1)</b>	PostMonth is the 1 month after the beginning of the current quarter (MM/YYYY).

You can also define a parameter with another parameter, adding or subtracting months and days as needed. For example:

<b>ToDate=\$FromDate\$(d,+7)</b>	ToDate is 7 days after FromDate (MM/DD/YYYY).
<b>PostMonth=\$ToDate\$(m,-1)</b>	PostMonth is 1 month before ToDate (MM/YYYY).

### Limitations of Task Runner

You cannot generate a YSR report through Task Runner if the report requires a digital signature at run-time. For more information, see “Options for digitally signing YSR reports” on page 144.

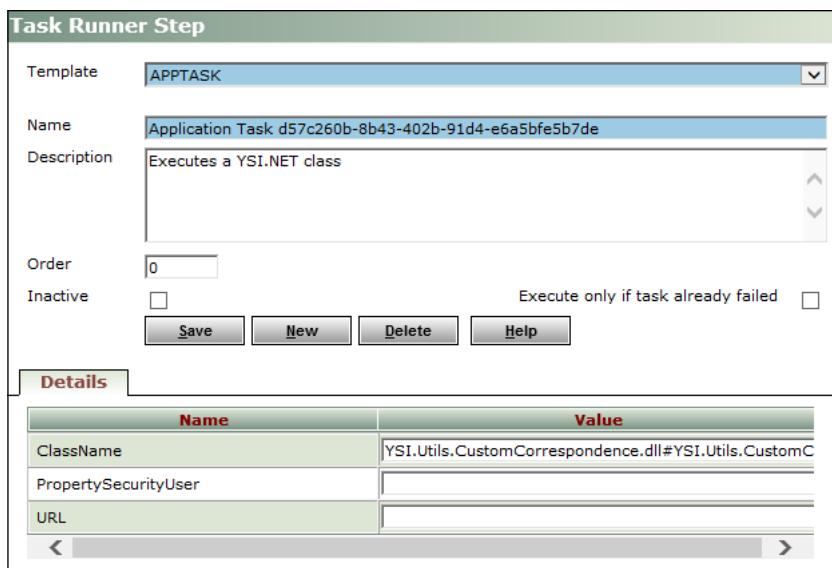
### To generate YSR reports using Task Runner

- 1 Select **Admin > Task Runner > Task > Add**. The **Task Runner Task** screen appears.

Task Runner Task	
Code	YSR
Name	YSR reports
Description	YSR monthly report packet
<input type="button" value="Save"/> <input type="button" value="New"/> <input type="button" value="Delete"/> <input type="button" value="Help"/>	

- 2 Complete the screen and click **Save**. The **Steps** tab appears.

- 3 On the **Steps** tab, click the **New Record** button  . The **Task Runner Step** screen appears.



Name	Value
ClassName	YSI.Utils.CustomCorrespondence.dll#YSI.Utils.CustomC
PropertySecurityUser	
URL	

- 4 Complete the fields.



Use the following class name:  
YSI.Utils.CustomCorrespondence.dll#YSI.Utils.CustomCorrespondence.AppClasses.ysiYSRCorrespondenceTask



In the **URL** field, you can enter the full URL, or you can point Voyager to a text file in your reports path that contains the URL. To point Voyager to the Reports path, enter: URLTemplate=filename.txt (substitute the filename of your text file).

- 5 Click **Save**.

## Report Scheduler

### In this section:

Preparing a Report Scheduler Script .....	152
Example Report Scheduler Script .....	153
Scheduling a YSR Report that Lacks a Top-Level Select .....	154
Scheduling Your Report .....	155
Monitoring Reports Generated with Report Scheduler .....	157

This section describes how to schedule YSR reports in advance in Report Scheduler. Report Scheduler is a component of Service Manager, a Plug-in that supplies utilities to Voyager. It relies on a set of scripting conventions that are unique to Yardi's scripting engine. This section shows how to write a script using the supported conventions such that Report Scheduler executes a YSR report.

You can also set up reports so that users can generate reports on the fly in Report Scheduler. For information about the latter approach, see “Generating Ad Hoc Reports in Report Scheduler” on page 137.

## Preparing a Report Scheduler Script

To schedule your report, you must prepare a script like the one in the following graphic. The script must begin with the prefix **rs\_**, and you must save the script to your Reports path. See the grid below the graphic for commentary about each part of the script.



The following graphic is cut off on the right-side for lack of space. For the full text file in a format that you can copy and paste, see “Example Report Scheduler Script” on page 153.

```
rs_YSR_UnitDir - Notepad
File Edit Format View Help
//Vista

//Database
analytic YSI._Utils.CustomCorrespondence#YSI.Utils.CustomCorrespondence.ysiYSRConductorI
//End Database

//Title
YSR Unit Directory
//End Title

//Select
select distinct sCode from property p where 1 = 1
#CONDITIONS#

//End Select

//Columns
//Type Name Head1 Head2 Head3 Head4 Show Color
T, , , , Property, Y, ,
//End columns

//Filter
//Type,DataType, Name, Caption, Key, List, Val1,Val2, Mandato
C, I, Property, Property, N, 524, p.hmy = #Property#, ,
C, T, ReportCode, ReportCode, N, "Select 'UnitDir'", ,
//End Filter
```

<b>Filename</b> <b>(rs_YSR_UnitDir.txt)</b>	You must save the script as a text file, using the prefix <b>rs_</b> in the filename. Save the file to your Reports path.
<b>//Vista</b>	A convention unique to the Yardi scripting engine. Not required for Report Scheduler but it is a good practice to preserve the classic components of a Yardi script.
<b>//Database</b>	Required. Your script must contain the //Database...//END Database section, and it must state the Correspondence class name exactly as it appears here. Copy and paste from the “Example Report Scheduler Script” on page 153.
<b>//Title</b>	Required. The title defined here appears on setup screens in Report Scheduler. <b>TIP</b> Keep your title short.

---

//Exclude	Optional. Prevents the YSR report from appearing in the SQL Reports list.
//Select	Required. The best practice is to use a very simple top-level select statement in your YSR report and use the same top-level select statement here. If your report does not contain a top-level select statement, use a simple select like the one illustrated here.  <b>NOTE</b> If your report does not have a top-level select statement, you must add a row to the //Filter section to force Report Scheduler to merge report data. For more information, see "Example Report Scheduler Script" on page 153.
//Columns	Required. The row count of this section must match the column count of the Select section.
//Filter Section	Critical. This section must recreate, at minimum, all the mandatory filter fields in the YSR report filter. It must also include a filter field that is hard-coded to hand off the YSR report code to the YSR report engine. You accomplish this by adding a row with the following <b>List</b> value:  <b>"Select 'YSRReportCode'"</b> where YSRReportCode stands in for the code of your report. For example, a report with code UnitDir requires this list value: <b>"Select 'UnitDir'"</b> .  <b>TIP</b> If your YSR report lacks a top-level select statement, you must also add a filter field with <b>Name</b> = Merge and <b>List</b> = "Yes". The purposes of the field is to hard-code the report so that Report Scheduler merges the data generated by YSR. For more information, see "Example Report Scheduler Script" on page 153.
	For more information about scripting the //Filter section, see the <i>Yardi SQL Scripting User Guide</i> .

---

## Example Report Scheduler Script

This section provides an example script that you can copy and paste into a text editor and adapt for your own reports. This is the same script as the script in the graphic in the preceding section.



You can also use the Yardi Excel Add-In to create this script for you. For more information, see "Dump SQL Tab Screen Reference" on page 275.

```
//Vista
//Database
analytic YSI._Utils.CustomCorrespondence#YSI._Utils.CustomCorrespondence.yсиYSRConductorInterface
//End Database

//Title
YSR Unit Directory
//End Title

//Select
select distinct sCode from property p where 1 = 1
#CONDITIONS#
//End Select

//Columns
//Type,Name,Head1,Head2,Head3,Head4      ,Show,Color,Formula,Drill,Key,Width,
T      ,      ,      ,      ,Property,Y      ,      ,      ,3      ,      ,500      ,
//End columns

//FILTER
//TYPE,Data,Name      ,Caption      ,Key,List      ,Val1      ,Val2,Mand,Multi,Title,
C      ,T      ,Property      ,Property      ,N      ,524      ,p.hmy = #Property#,      ,Y      ,Y      ,Y      ,
C      ,T      ,ReportCode,ReportCode,N      ,"Select 'UnitDir'",      ,      ,Y      ,Y      ,N      ,
```

```
//END FILTER
```

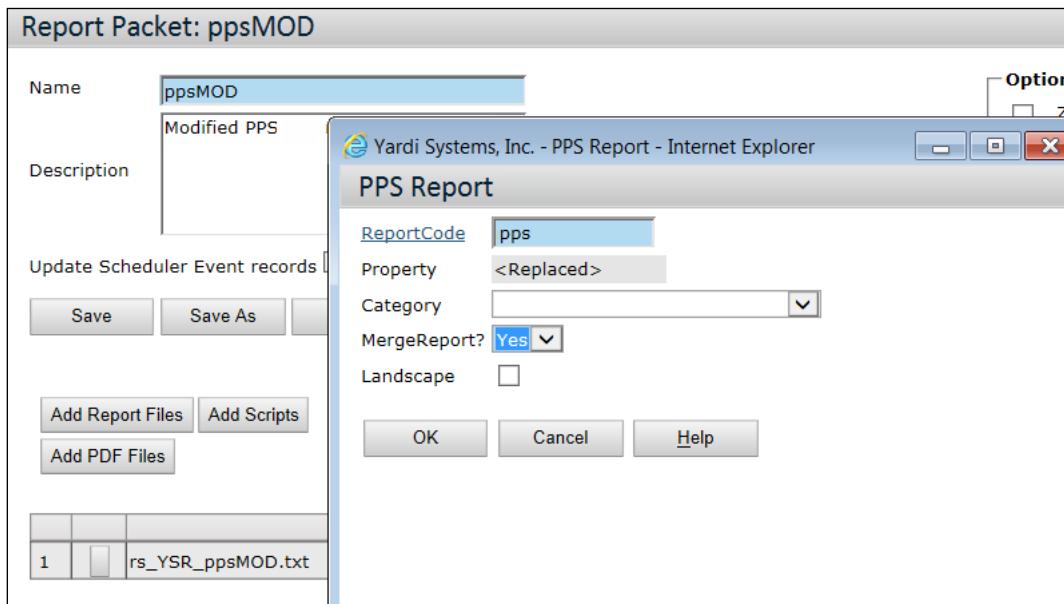
## Scheduling a YSR Report that Lacks a Top-Level Select

If your YSR report does not have a top-level select statement, or if problems persist even when using a top-level select statement, do the following:

- 1 To the //FILTER section of the Report Scheduler script, add a filter named Merge and set its List value to “Yes”.

```
//Filter
//Type, DataTyp,      Name,      Caption, Key,      List,      Val1
Title
C, T, ReportCode, ReportCode, N, "Select 'pps'", ,
C, T, hprop, Property, , 61, P.hMy = #hprop#,
L, T, Merge, MergeReport?, N, "Yes", , Y, N, N, N,
//end filter
```

- 2 In Report Scheduler, when setting up the report packet, select **Yes** in the **Merge** filter field.



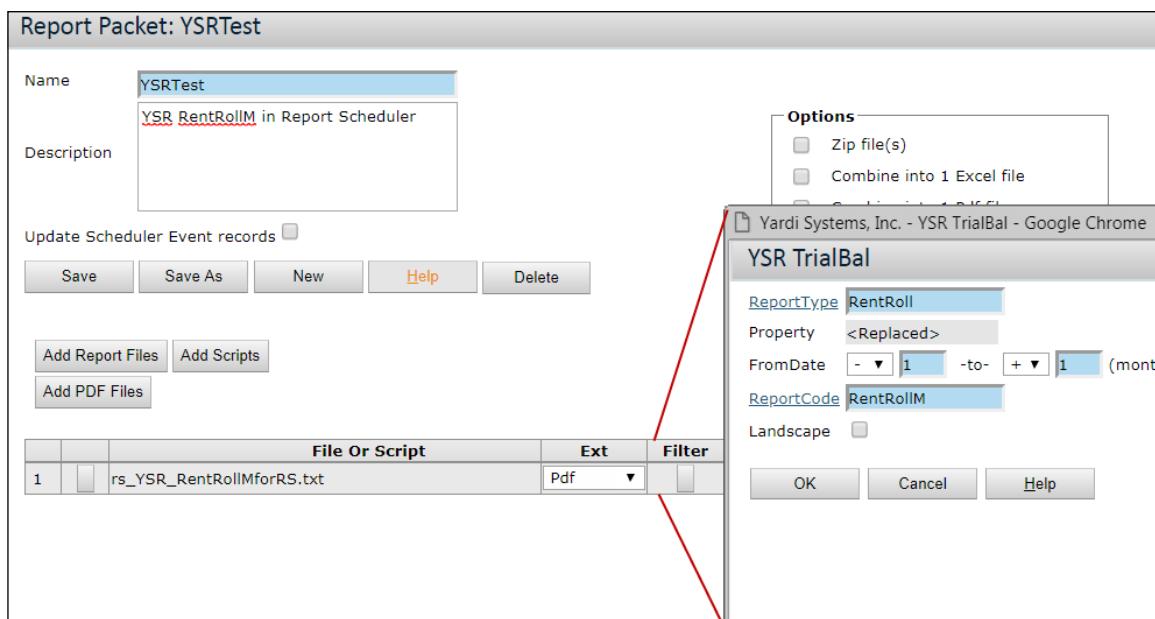
## Scheduling Your Report

After you save the file to the Reports path with the correct naming conventions, the file becomes accessible for scheduling using normal Report Scheduler procedures. This section provides tips and best practices relevant to YSR. For full documentation of Report Scheduler, see the *Service Manager Administration and Setup Guide*.

### To schedule a report in Report Scheduler

- 1 Add a report packet.

From System Administration: **Admin > Report Scheduler > Add a Report Packet**



This is where you define all filter criteria that you want to pass into the underlying YSR report, except for the property code. The property code is associated with the distribution member in the next step.



The + and - selectors (applicable for date fields) refer to the system date, not a financial post month.



If you edit your Report Scheduler script, it is strongly recommended to add a new report packet.

- 2** Add a distribution member. The distribution member is equivalent to the Voyager user that executes the report. It also defines the property or properties for which the report is generated.

From System Administration: **Admin > Report Scheduler > Add Distribution Member**

Distribution Member

Code	pm	Active	<input checked="" type="checkbox"/>
Last Name	Snipes	Notes	
First / Dear	Jason		
Address			
City			
State / Zip			
Office			
Home			
Fax			
Email			
Company			

**Save** **New** **Delete** **Help**

**Warning:** This Member is linked to at least one Event - changing data may affect future dates

**Property**

Packet per Property

Code	Description	Consolidate	Count
resny02	Empire Terrance	<input type="checkbox"/>	1
		<input type="checkbox"/>	

**Report Packets**

- ACH Pmt
- Owner Reports
- PHA Landlord/Vendor ACH Report Packet
- PHA Tenant ACH Report Packet
- ppsMOD
- Property Marketing Materials
- TenStmt
- TenStmt
- Unit Directory
- YSRTest

---

**Active** Select this check box.

**Property Grid** Enter the property or properties for which you want to generate the report.

**TIP** If you want to generate a consolidated report, enter a property list in the **Code** column and select the **Consolidate** button before you click **Save**. If you enter a property list and click **Save** before you select **Consolidate**, Voyager splits the property list into its component properties and you cannot recover the option to consolidate. You must delete the record and start over.

- 
- 3** Schedule the report.

## Monitoring Reports Generated with Report Scheduler

You can monitor the user activity of reports generated with Report Scheduler. Voyager stores these report requests in the Request\_QUE and the Request \_QUEURL tables. When you want to track generation activity, Voyager gathers the data from these tables that are associated with Voyager users in the PMUser, Person, and PREmployee tables to show you all Service Manager Request\_Queue submissions for the user as well as the report status for the date range that you specify.

The screenshot shows the 'YSR Monitor Report Requests' interface. At the top, there are filter fields for 'User' (set to 'Jonl'), 'Status' (dropdown), 'Scheduled Date' (set to '01/01/2018' to '12/03/2018'), and a checkbox for 'Show Only YSR Reports'. Below these are 'Submit' and 'Clear' buttons. The main area displays a table of report requests:

Delete	Wait	User	Report Name	Status	Scheduled	Started	Completed	Report Link	Report Link with No Email Id	Email To	Output Type	Result
		jonl		Processed	01/27/2018 09:01:11 AM	01/28/2018 09:01:14 AM	01/28/2018 09:01:31 AM	<a href="#">View Report</a>				Successful
		jonl	PoolCheckInHist	Processed	01/27/2018 08:57:19 AM	01/28/2018 08:57:24 AM	01/28/2018 08:57:31 AM	<a href="#">View Report</a>				Successful
		jonl	YSRLeaseDisplay	Processed	01/27/2018 08:51:52 AM	01/28/2018 08:51:54 AM	01/28/2018 08:52:14 AM	<a href="#">View Report</a>				Successful
		jonl	YSRRentRoll	Processed	01/24/2018 10:56:59 AM	01/25/2018 10:57:01 AM	01/25/2018 10:57:56 AM	<a href="#">View Report</a>				Successful

### To monitor reports generated with Report Scheduler

- From the **YSR Admin** side menu, select **Reports > YSR Monitor Reports Requests**. The **YSR Monitor Report Requests** screen appears.



You can also access the **YSR Monitor Report Requests** screen by adding a custom side menu to this page location:  
`../pages/CustomCorrespReportMonitor.aspx`

- Complete the filter. For field descriptions, see "YSR Monitor Report Request Screen Reference" on page 157.
- Click **Submit**.

### YSR Monitor Report Request Screen Reference

The screenshot shows the 'YSR Monitor Report Requests' interface. It includes filter fields for 'User' (empty), 'Status' (dropdown), 'Scheduled Date' (set to '12/02/2018' to '12/03/2018'), and a checkbox for 'Show Only YSR Reports'. Below these are 'Submit' and 'Clear' buttons.

User

Filter by Voyager user, or leave the field blank to see Report Requests for all users.

<b>Status</b>	Indicates the progress status of the report as reported by Service Manager:  <b>Not Processed</b> Service Manager has not collected the report for processing, or the report is scheduled in the future.  <b>In Process</b> Service Manager is actively working on the report.  <b>Processed</b> Service Manager has successfully generated the report.  <b>Problem</b> Service Manager completed the procedure, but the report did not generate successfully.
<b>Scheduled Date</b>	Date range in which the report was scheduled to be processed in Service Manager. The <b>Date To</b> extends to 11:59PM on the day that you specify.  <b>NOTE</b> The <b>Scheduled Date From</b> and <b>To</b> fields default to yesterday and today.  <b>CAUTION</b> For large databases, it is best to select shorter date ranges as larger date ranges involve significant processing that can slow server performance for other users.
<b>Show Only YSR Reports</b>	Shows only request queue reports identified as YSR reports. If left unchecked, all report requests are included.

## Side Menus

### In this section:

Adding Custom Menu Paths to YSR Reports in Voyager.....	158
Automatically Generating YSR Reports from a Side Menu .....	160
Adding Menu Paths for YSR Reports in VoyagerPlus .....	161
Adding the YSR Admin Menu.....	162

This section describes how to create custom side menu paths for YSR reports.

## Adding Custom Menu Paths to YSR Reports in Voyager

You can create a custom side menu that directs users to one or more of your YSR reports. This helps your users find and generate YSR reports more easily.

When you create the custom side menu, you add a custom URL that Voyager uses to direct users to a screen with the options that you specify. You can set up your custom URL to direct users to a specific YSR report, or you can give users options to choose between multiple reports. This section provides an example of the latter option, in which users can choose between multiple reports. To restrict users to just one report, include only one report code in the report code parameter of your custom URL.

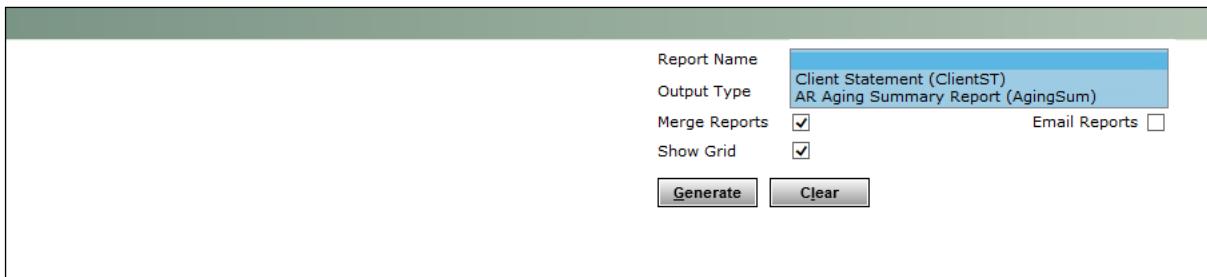
### Example URL for Custom Side Menu

./pages/CustomCorrespGenerate.aspx?ReportCode=ClientST,AgingSum  
&OutputType=PDF,Excel&Merge=True&Grid=True

## URL Components

Parameter	Parameter Value	Mandatory	Example
ReportCode=	The code of the report you want users to generate. To enable users to choose between specific reports, enter each report code separated by a comma or a caret (^).	Yes	ClientST,AgingSum
&OutputType=	PDF, Excel, or Screen. The format in which Voyager generates the report. If you want to enable users to choose between output types, enter all applicable report types separated by a comma or a caret (^).  <b>TIP</b> The first value in the URL is the default. For example, if the URL includes &OutputType=word,PDF, then Word is the default output type.		PDF,Excel
&Merge=	True or False. If &Merge=True, Voyager merges the reports into a single file.		True
&Grid=	True or False. If &Grid=True, Voyager shows grid lines for on screen reports.		True
&HideFilter=	True or False. If true, Voyager does not display the YSR report filter fields. Use this option if you hard-code your filter values and you do not want users to apply filtration. Users can generate the report but cannot filter for different search criteria.		False
&Generate=	True or False. If true, Voyager generates the report automatically.  <b>TIP</b> This parameter is applicable only for menu paths that direct the user to a single YSR report. When the parameter is set to True, Voyager generates the report automatically.  <b>NOTE</b> If you set this parameter to True, you must define all mandatory filter items in the URL and you must define one (and only one) output type.  For more information, see “Automatically Generating YSR Reports from a Side Menu” on page 160.		True
&FilterFieldName=	Acceptable values depend on the data type the filter field accepts.  You can pre-populate YSR report filter fields by including in the URL a parameter that refers to a filter field by name. For example, if you have a filter field named <b>Property</b> and it uses a YSI property lookup, you can include the following parameter in your URL:  &Property=scode		

The custom URL detailed in the preceding sections creates a YSR report generation screen. On this screen, users can choose between a Client Statement report (ClientST) and an AR Aging report (AgingSum), as depicted here:



### To add a custom side menu for YSR reports

- 1 From Voyager System Administration, select **Security > Add Menus**. The **Create New Menuset** screen appears.
- 2 Complete the required fields and click **OK**. The **Voyager Menu Editor** screen appears.

Caption	Link
"YSR Report"	..pages/CustomCorrespGenerate.aspx?ReportCode=gClientST,ARAGing &OutputType=PDF,Excel

- 3 Add a caption and enter the custom URL for your YSR report in the **Link** field.
- 4 Click **Save**.

## Automatically Generating YSR Reports from a Side Menu

You can set up a custom side menu path that automatically generates a YSR report. When the user follows the menu path, YSR report data appears immediately; the user does not have to supply filter criteria or click **Generate**.

To add a menu path that automatically generates a YSR report:

- 1 Add the following parameter to your side menu path URL: **&Generate=True**
- 2 Include and define all mandatory filter items as parameters in the URL. (Use filter field names as parameters.)
- 3 Set the output type parameter to one type only (PDF, Excel, or Screen).

#### Example

`./Pages/CustomCorrespGenerate.aspx?ReportCode=RentRoll&PropertyList=comoff01&FromDate=02/20/2007&OutputType=Screen&Merge=True&Grid=True&Generate=True`

## Adding Menu Paths for YSR Reports in VoyagerPlus

You can add links to specific YSR reports or the YSR report generation screen in VoyagerPlus.

To add menu links in VoyagerPlus, edit the VoyagerPlus menu set in Voyager's menu editor. The following graphic shows two links to specific YSR reports followed by a link to the YSR report generation screen (where users can select from all YSR reports that are configured to display as VoyagerPlus reports).

- "YSR Reports"	
-- "Packages Delivered not Picked Up"	"ysr/?ReportID=84"
-- "Packages Pickup By Unit or Resident"	"ysr/?ReportID=83"
-- "Pick a YSR Report"	"ysr/"

To create the links to specific YSR reports in older VoyagerPlus releases, you must know the ReportID of your YSR report.



In later VoyagerPlus releases, you only do not need to know the report ID. Instead, you only need your YSR report code (for example, "ysr/?ReportCodes=MyYSRReportCode").

The ReportID is the hmy of the report in the CustomCorrespMerging table (CustomCorrespMerging.hmy).

To retrieve ReportIDs, you can use ySQL to process this query:

```

SELECT
CM.hmy as ReportID,
cm.sCode as ReportCode,
cm.sName as ReportName
FROM CustomCorrespMerging CM
Order by 2

```



If you still cannot access the hmy of your YSR report, contact Yardi technical support.

You can add parameters to the menu items to configure the linked YSR report in specific ways. For example, the following string creates a link to a YSR report that is ready for emailing as an attachment:

"ysr/?ReportID=84&isAttachReports=true&isEmailReports=true"

The following parameters are supported:

---

**&selOutputType=undefined**

---

**&isAttachReports=false**

---

**&isMergeReports=false**

---

**&isShowGrid=false**

---

**&isPublishReport=false**

---

**&isShowOnPortal=false**

---

**&Attach=-1** Where -1 is true, if the report has attachment setup.

---

**&Email=-1** Where -1 is true, if the report has email and attachment setup and user has access to email (MyBase.AccessType"CustomcorrespGenerate\_AllowEmail").

---

**&PublishtoSharePoint=-1** Where -1 is true, if the report has attachment setup.

---

## Adding the YSR Admin Menu

Report designers and administrators can access screens to set up YSR reports and signatures, as well as produce, deliver, and monitor YSR reports from the YSR Admin menu.

To add the YSR Admin menu set, load the following package: Opt\_ss\_YSRAdmin\_MenuSet.pkg.



## Custom Help

You can configure individual reports so that, when the users clicks the **Help** button on the **Report Generation** screen, they are directed to custom, module-specific online Help topics. The user must click on the **Help** button prior to clicking the **Generate** button to access module-specific content. If the user generates the report before accessing the online Help, the **Help** button redirects to the default online YSR Help material.

This approach requires coordination with other Voyager product teams; custom Help is available on a limited, case-by-case basis. Contact Yardi technical support for more information.

If custom Help content is available from another Voyager product, the product team supplies the YSR report writer with a token. The YSR report writer links the YSR report to the Help by defining the token on the **Output Options** tab, as highlighted in the following graphic.

Help Topic	PHAYSR

## CHAPTER 6

# Custom Financial Analytics Reports in YSR

### In this chapter:

CFA Report Setup Overview.....	165
CFA Account Tree Reports in YSR .....	170
CFA Property Portfolio Reports in YSR.....	181
CFA Comparison Reports in YSR .....	187

If your organization uses Custom Financial Analytics to generate custom reports, you can adapt your CFA reports for use with YSR.

There are several benefits to using CFA reports in YSR. If you already have a complex report designed in CFA, you do not have to recreate the whole design in YSR. You can simply point YSR to the CFA report template and retrieve the CFA report data that you have already configured. YSR also offers far greater report formatting and customization options than CFA. With YSR, you can format your report templates any way you like, create charts, graphs, and pivot tables (YSR), and include your CFA reports in more complex YSR report packets.

CFA reports also have the advantage that they derive their data from Financial Analytics. You can of course base a YSR report on Financial Analytics directly, but by using CFA instead, you benefit from a more straightforward interface with YSR. You can also break the report design process into two parts: design and verify the data in CFA, and format your report in YSR by manipulating your report template.

### CFA report types

This section describes how to adapt each type of Custom Financial Analytics (CFA) report for use in YSR. CFA report types include:

- Account Tree reports
- Property Portfolio reports
- Property Comparison reports
- Attribute Comparison reports

Because each report type retrieves, aggregates, and displays data in different ways, each report type has different setup requirements. For background information about each type of CFA report, see the *Custom Financial Analytics Reporting User's Guide*.

## CFA Report Setup Overview

### In this section:

Setting Up CFA Reports in YSR.....	165
Generating Consolidated Data .....	167
Formatting CFA Reports.....	169

This section provides an overview of setup tasks common to all types of Custom Financial Analytics (CFA) reports when used with YSR.

### Setting Up CFA Reports in YSR

To adapt a CFA report for use in YSR, you must reproduce the following elements of the CFA report:

- The rules for retrieving data, encapsulated in the CFA report template
- The filter fields the user completes at run time
- The design of the report output

## Rules for Retrieving Data (CFA Report Template)

To use a CFA report template to retrieve data for a YSR report, you must provide the CFA template code when setting up your YSR report sections. Simply define the template code as the constant value for the **Template** filter element of the CF data source, as illustrated here:

The screenshot shows the 'Report Sections Setup' dialog box. At the top, it displays the YSR Report: 'CFA Balance Sheet (cfabs)' and the Report Code: 'cfabs'. Below this are 'Save', 'Close', and 'Help' buttons.

The main area is titled 'Report Sections' and contains a table with columns: Section Code, Description, SELECT Name, Standard Analytics Report, Map Standard Analytics Filter, Tokens, Relations, and Key Columns. A row for 'CF' (Custom Financials) is selected, and a sub-dialog box titled 'Map Standard Analytics Filter' is overlaid on the table. This sub-dialog shows the YSR Report: 'CFA Balance Sheet (cfabs)', Report Code: 'cfabs', and SectionCode: 'CF'. It also has 'Save', 'Close', and 'Help' buttons. The 'Map Standard Analytics Filter' table has columns: Standard Analytics Report Filters, YSR Report Filters, and Constant Value. It lists five filter fields: Template (selected), PropertyCode, FromMMYY, ToMMYY, and SuppressZero. The Constant Value column contains 'CFA\_BS' for the Template field and '1' for the SuppressZero field.

## Filter Fields for CFA Reports in YSR

Every CFA report has slightly different filter requirements. Some CFA reports require the user to select from different account trees or G/L books; some do not. When setting up your YSR report to generate a CFA report, therefore, add only the top-level filter fields required by the CFA and YSR report designs. Then map the top-level filter fields to the filter elements associated with Custom Financials. (Complete filter mapping on the **Report Sections Setup** screen.)

## Report Output Design

When you adapt a CFA report for use in YSR, you must recreate the design of the report output. That is, you must create a new report template, either in Excel or Word.

All CFA data is organized in numbered columns (column1, column2, column3, etc.). Use numbered columns as the field names for Excel smart markers and Word merge fields.

Example smart markers include:

&=[sectioncode].[column1]

&=[sectioncode].[column2]

&=[sectioncode].[column3]

## Generating Consolidated Data

If you want your report to generate consolidated data (like aggregate GL data for prop1^prop2), you must manipulate the **IsConsolidate** filter element native to Custom Financials. You can either hard-code the **IsConsolidate** element by assigning it a constant value of **True** or **False**, or you can set up a top-level filter check box for users to complete at run time (you must map the top-level filter to the **IsConsolidate** element).

If you want users to be able to turn consolidation on and off, complete the following tasks:

- 1 Create a filter field (a check box) where users can indicate that they want consolidated data.
- 2 Map the filter field to the **IsConsolidate** filter element native to Custom Financials.

For example, the following graphic shows the custom filter fields associated with a YSR report that generates CFA data.

CFA Balance Sheet		Report Settings		
Property	comoff01^comind01	Report Name	CFA Balance Sheet (cfabs)	
From Date	01/2007	Output Type	Screen	<input type="checkbox"/> Attach Reports
To Date	12/2007	Merge Reports	<input type="checkbox"/>	<input type="checkbox"/> Email Reports
Consolidate	<input checked="" type="checkbox"/>	Show Grid	<input type="checkbox"/>	<input type="checkbox"/> Show on Portal
		<input type="button" value="Generate"/> <input type="button" value="Clear"/>		

The **Consolidate** check box is mapped to the **IsConsolidate** filter element, as depicted below.

Standard Analytics Report Filters	YSR Report Filters	Constant Value
Template		CFA_BS
PropertyCode	Property	
FromMMYY	FromDate	
ToMMYY	ToDate	
SuppressZero		1
IsConsolidate	Consolidate	<input checked="" type="checkbox"/>

## Adding Top-Level Selects for Consolidated Reports

When you give users the option to consolidate data or break out data by property, you must adjust your top-level select statement. To adequately represent consolidated data, the top-level select statement must return just one row of data. To represent report data broken out by property, by contrast, it must return multiple lines of data.

To accommodate this requirement:

- Add to your report a top-level filter field, using the check box data type. Note the field name, as you must refer to the field name in the top-level select statement. In this example the check box has field name **IsConsolidate**).
- Map your top-level check box to the **IsConsolidate** filter element native to Custom Financial Analytics (as per normal setup procedures).
- For your top-level select statement, start by authoring two select statements that return the same columns. These are the columns you want to see on the report generation screen at run time.
- To one of the selects, add a WHERE clause that includes this string:



This select returns data when the user does not consolidate data.

- To the other select, add a WHERE clause that includes this string:



This select returns data when the user consolidates. Author it so that it returns just one line of data.

- Join the two selects with a UNION ALL operator.

## Example script

An example script follows. Adapt it as necessary for your own reports.

```

SELECT
    hMy PropertyId,
    sCode PropertyCode,
    sAddr1 PropertyName
  from property
where
    hmy in
    (#property#)
    AND UPPER(''#IsConsolidate#'') = '0'

Union ALL

SELECT
    -1 PropertyId,
    'Consolidated Properties' PropertyCode,
    'N/A' PropertyName
WHERE
    UPPER(''#IsConsolidate#'') = '1'

```

## Formatting CFA Reports

This section discusses formatting options for CFA reports in YSR.

### Number formatting

Apply number formatting in your Excel report template.

The Custom Financials data source does expose the **DecimalDigits** field element to YSR (visible on the filter mapping screen for the Custom Financials data source) but it is not supported.

### Bold, underline, italics, and other formatting

When you select CFA as the data source for a YSR report, CFA exposes to YSR a range of data related to formatting. You can access that data by creating smart markers for each formatting flag in CFA. For each formatting flag, Voyager returns a TRUE or FALSE (or 0, 1, or 2, in the case of underlining).

You can take advantage of this information by using conditional formatting in your Excel report template. Add conditional formatting rules so that, for example, Excel applies bold formatting in the row where the bBold smart marker returns TRUE. For an example, see “Account Tree Report with Bold and Underline” on page 170.

## CFA formatting field names and values

Field Name	Values	Supported in YSR
<b>bBold</b>	TRUE	YES
	FALSE	
<b>iUnderline</b>	0 (no underlining)	YES
	1 (single underline)	
	2 (double underline)	
<b>iIndent</b>	TRUE	NO
	FALSE	
<b>bItalic</b>	TRUE	YES
	FALSE	
<b>bNegate</b>	TRUE	NO
	FALSE	
<b>iType</b>	TRUE	NO
	FALSE	
<b>iAfter</b>	TRUE	NO
	FALSE	
<b>iNest</b>	TRUE	NO
	FALSE	

## CFA Account Tree Reports in YSR

### In this section:

Account Tree Report with Bold and Underline .....	170
Account Tree Report with Manipulated Data .....	176

This section provides examples of CFA Account Tree reports adapted for use in YSR. Account Tree reports display GL data for one or more properties in a single column. (Comparison reports, by contrast, display data in multiple columns depending on user input.) Because report data appears in just one data column, report design is relatively simple. No additional scripting is necessary.

### Account Tree Report with Bold and Underline

This section describes how to create a basic account tree CFA report in YSR. The example makes some simplifications, but provides the necessary framework for any account tree report.

## To set up a CFA account tree report for YSR

- Review your CFA report in Voyager.

**Custom Financial Reports**

Property	comoff01	Asset Manager		Denominator	
Book	Accrual	Assistant Manager		Department	
Report Template	CFA_BS	Client		Source	
Period	12/2007	Country		Immediate Source	
Summary By	Client	District Manager		DecimalDigits	0
Account Tree	mod_bs	Fund Objective		Suppress Zero	
Freeze Pane <input type="checkbox"/>					
		Show Account <input checked="" type="checkbox"/> ShowColumn <input checked="" type="checkbox"/> Detail <input type="checkbox"/> Summary <input type="checkbox"/> Tree Level <input checked="" type="checkbox"/> Grid			
		Value			

**1000-0000 ASSETS**

<b>1001-0000 CURRENT ASSETS</b>	
1100-0000 CASH & CASH EQUIVALENT	
1110-0000 Cash - Operating	482,208
<b>1190-0000 TOTAL CASH &amp; CASH EQUIVALENT</b>	<b>482,208</b>
<b>1499-0000 TOTAL CURRENT ASSETS</b>	<b>482,208</b>
<b>1500-0000 FIXED ASSETS</b>	
1501-0000 PROPERTY	
1600-0000 ACCUMULATED DEPRECIATION	
1605-0000 A/D Buildings	-50,000
1610-0000 A/D Exterior	-155
1615-0000 A/D Roof	-722
1620-0000 A/D Common Area	-145
1625-0000 A/D Elevator Upgrades	-158
1635-0000 A/D Parking Lots	-122

- Review your CFA report design.

**Custom Financial Reports Layout**

Code	CFA_BS	Type	Account Tree	Notes																																								
Name	Custom Financial Balance Sheet																																											
Account Tree	mod_bs	Report For Invest. Mngt. Module <input type="checkbox"/>																																										
Columns	1	Apply User Group Security <input type="checkbox"/>																																										
<input type="button" value="Edit"/> <input type="button" value="New"/> <input type="button" value="Close"/> <input type="button" value="Help"/> <input type="button" value="Delete"/> <input type="button" value="Save As"/> <input type="button" value="Set Permissions"/>																																												
<b>Financial Formulas</b> <table border="1"> <tr> <td>New Row</td> <td>Function</td> <td>Copy Columns</td> <td>From</td> <td>Column 1</td> <td>to</td> <td>Column 1</td> <td>Execute</td> </tr> <tr> <td><input type="button" value=""/></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>ID</td> <td>From Acct</td> <td>To Acct</td> <td>Column 1</td> <td></td> <td>Delete</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td></td> <td></td> <td>Value</td> <td><input type="button" value=""/></td> <td><input type="button" value=""/></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td></td> <td></td> <td>MTD</td> <td><input type="button" value=""/></td> <td><input type="button" value=""/></td> <td></td> <td></td> </tr> </table>					New Row	Function	Copy Columns	From	Column 1	to	Column 1	Execute	<input type="button" value=""/>								ID	From Acct	To Acct	Column 1		Delete			1			Value	<input type="button" value=""/>	<input type="button" value=""/>			2			MTD	<input type="button" value=""/>	<input type="button" value=""/>		
New Row	Function	Copy Columns	From	Column 1	to	Column 1	Execute																																					
<input type="button" value=""/>																																												
ID	From Acct	To Acct	Column 1		Delete																																							
1			Value	<input type="button" value=""/>	<input type="button" value=""/>																																							
2			MTD	<input type="button" value=""/>	<input type="button" value=""/>																																							

**3 Prepare your YSR report template in Excel.**

- a Add smart markers, including smart markers that retrieve formatting information (columns D and E).

A	B	C	D	E
1				
2				
3	<b>Account Name</b>	<b>Account Code</b>	<b>Value</b>	<b>bBold</b>
4	&=[CF].[acctDesc]	&=[CF].[acctcode]	&=[CF].[column1]	&=[CF].[bBold]
5				&=[CF].[iUnderline]

- b Use conditional formatting to apply bold formatting when column D has a value of TRUE.

The screenshot shows a Microsoft Excel spreadsheet with a table structure. The columns are labeled A through G. Rows 1 and 2 are blank. Row 3 contains headers: Account Name, Account Code, Value, bBold, and iUnderline. Row 4 contains formulas: &=[CF].[acctDesc], &=[CF].[acctcode], &=[CF].[column1], &=[CF].[bBold], and &=[CF].[iUnderline]. Row 5 is blank. A 'New Formatting Rule' dialog box is open over the table, centered on row 4. The dialog box has a title bar 'New Formatting Rule' and a close button. It contains a section 'Select a Rule Type:' with several options listed as bullet points. Below that is a section 'Edit the Rule Description:' with a sub-section 'Format values where this formula is true:' containing the formula =D4=TRUE. There is a preview window showing the text AaBbCcYyZz in bold. At the bottom right of the dialog box are 'OK' and 'Cancel' buttons.



Excel prompts you to pick a specific cell like **\$D\$4**. This fixes the conditional formatting rule in place, but you need the rule to apply to each dynamically expanding row. Remove the \$ before the row number (**\$D4**) so that the conditional formatting expands dynamically when you generate the report.

- c Use conditional formatting to apply underlining to your report.



Some versions of Excel do not support double underlining with conditional formatting. In that case, first add double underlining to your report template and then use conditional formatting to remove the double underlining.

The following graphic shows the **Conditional Formatting Rules Manager** window after all rules are applied (two rules for underlining and one rule for bold).

The screenshot shows the Conditional Formatting Rules Manager dialog box overlaid on an Excel spreadsheet. The dialog box has the following interface:

- Show formatting rules for:** Current Selection
- Buttons:** New Rule..., Edit Rule..., Delete Rule, Up/Down arrows, OK, Close, Apply.
- Table:**| Rule (applied in order shown) | Format | Applies to | Stop If True |
| --- | --- | --- | --- |
| Formula: =\$E4=2 | AaBbCcYyZz | =A\$4:\$C\$4 |  |
| Formula: =\$D4=TRUE | **AaBbCcYyZz** | =A\$4:\$C\$4 |  |
| Formula: =\$E4=1 | AaBbCcYyZz | =A\$4:\$C\$4 |  |

- d Hide columns D and E.

	A	B	C	F
1				
2				
3	Account Name	Account Code	Value	
4	&=[CF].[acctDesc]	&=[CF].[acctcode]	&=[CF].[column1]	
5				

- e Save your Excel template to the Reports path.

**4 Add a YSR report.**

The screenshot shows the 'YSR Report Setup' interface. Key fields include:

- Code:** cfabs
- Description:** CFA Balance Sheet
- Top-Level Select Statement:** (empty)
- OR Script File:** (empty)
- Key Column:** (empty)
- Inactive:**
- Exclude From Generate List:**

Buttons at the bottom include: Save, New, Save Copy, Define Filters, Map Top-Level Filter, Dump SQL, Delete Setup, Help.

Below the main setup, a tab bar shows: Report Setup (selected), Attachment & Email, Additional Attachment & Email, Output Options, Digital Signature.

A detailed table lists report configurations:

Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?
1	cfabs	CFA_BS_template.xlsx			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**5 Add custom filter fields to your YSR report.**

The screenshot shows the 'Report Filters Setup' interface. It includes:

- YSR Report: CFA Balance Sheet (cfabs)
- Buttons: Save, Preview, Help.

A table lists custom filter fields:

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?
Property	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
FromDate	From Date	2	Post Month		<input type="checkbox"/>		<input type="checkbox"/>
ToDate	To Date	3	Post Month		<input type="checkbox"/>		<input type="checkbox"/>

**6 On the Report Sections Setup screen, select Custom Financials as your data source.**

The screenshot shows the 'Report Sections Setup' interface. It includes:

- YSR Report: CFA Balance Sheet (cfabs)
- Report Code: cfabs
- Buttons: Save, Close, Help.

Below the main setup, a tab bar shows: Report Sections (selected), Attachment Setup.

A table lists report sections:

Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns
CF	Custom Financials		Custom Financials	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

- 7 Map filters. Map the **Template** element in the **Standard Analytics Report Filters** column to the code of the CFA report you want to generate in YSR.

**Map Standard Analytics Filter**

YSR Report:	CFA Balance Sheet (cfabs)																								
Report Code:	cfabs																								
SectionCode:	CF																								
<input type="button" value="Save"/> <input type="button" value="Close"/> <input type="button" value="Help"/>																									
 <table border="1"> <thead> <tr> <th>Standard Analytics Report Filters</th> <th>YSR Report Filters</th> <th>Constant Value</th> </tr> </thead> <tbody> <tr> <td>Template</td> <td></td> <td>CFA_BS</td> </tr> <tr> <td>PropertyCode</td> <td>Property</td> <td></td> </tr> <tr> <td>FromMMYY</td> <td>FromDate</td> <td></td> </tr> <tr> <td>ToMMYY</td> <td>ToDate</td> <td></td> </tr> <tr> <td>SuppressZero</td> <td></td> <td>1</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>		Standard Analytics Report Filters	YSR Report Filters	Constant Value	Template		CFA_BS	PropertyCode	Property		FromMMYY	FromDate		ToMMYY	ToDate		SuppressZero		1						
Standard Analytics Report Filters	YSR Report Filters	Constant Value																							
Template		CFA_BS																							
PropertyCode	Property																								
FromMMYY	FromDate																								
ToMMYY	ToDate																								
SuppressZero		1																							



If your YSR report has a top-level **Book** filter field (supported by `ysiBookLookup`), map the **Book** filter field to the **BookCodeList** element in the **Standard Analytics Report Filters** column. If you fail to specify a book for the **BookListCode** element, CFA defaults to the Cash book.

- 8 Generate your YSR report.

**CFA Balance Sheet**

Property	comoff01	Report Name	CFA Balance Sheet (cfabs)																																																		
From Date	01/2007	Output Type	Screen																																																		
To Date	12/2007	Merge Reports	<input type="checkbox"/>																																																		
		Show Grid	<input type="checkbox"/>																																																		
		<input type="button" value="Generate"/>	<input type="button" value="Clear"/>																																																		
		Show on Portal <input type="checkbox"/>																																																			
<table border="1"> <thead> <tr> <th>Account Name</th> <th>Account Code</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td><b>ASSETS</b></td> <td><b>1000-0000</b></td> <td></td> </tr> <tr> <td><b>CURRENT ASSETS</b></td> <td><b>1001-0000</b></td> <td></td> </tr> <tr> <td>CASH &amp; CASH EQUIVALENT</td> <td>1100-0000</td> <td></td> </tr> <tr> <td>Cash - Operating</td> <td>1110-0000</td> <td>482,208.37</td> </tr> <tr> <td><b>TOTAL CASH &amp; CASH EQUIVALENT</b></td> <td><b>1190-0000</b></td> <td><b>482,208.37</b></td> </tr> <tr> <td><b>TOTAL CURRENT ASSETS</b></td> <td><b>1499-0000</b></td> <td><b>482,208.37</b></td> </tr> <tr> <td> FIXED ASSETS</td> <td>1500-0000</td> <td></td> </tr> <tr> <td> PROPERTY</td> <td>1501-0000</td> <td></td> </tr> <tr> <td>ACCUMULATED DEPRECIATION</td> <td>1600-0000</td> <td></td> </tr> <tr> <td>A/D Buildings</td> <td>1605-0000</td> <td>-50,000.00</td> </tr> <tr> <td>A/D Exterior</td> <td>1610-0000</td> <td>-155.00</td> </tr> <tr> <td>A/D Roof</td> <td>1615-0000</td> <td>-722.00</td> </tr> <tr> <td>A/D Common Area</td> <td>1620-0000</td> <td>-145.00</td> </tr> <tr> <td>A/D Elevator Upgrades</td> <td>1625-0000</td> <td>-158.00</td> </tr> <tr> <td>A/D Parking Lots</td> <td>1635-0000</td> <td>-122.00</td> </tr> <tr> <td> <b>TOTAL ACCUMULATED DEPRECIATION</b></td> <td><b>1690-0000</b></td> <td><b>-51,302.00</b></td> </tr> </tbody> </table>		Account Name	Account Code	Value	<b>ASSETS</b>	<b>1000-0000</b>		<b>CURRENT ASSETS</b>	<b>1001-0000</b>		CASH & CASH EQUIVALENT	1100-0000		Cash - Operating	1110-0000	482,208.37	<b>TOTAL CASH &amp; CASH EQUIVALENT</b>	<b>1190-0000</b>	<b>482,208.37</b>	<b>TOTAL CURRENT ASSETS</b>	<b>1499-0000</b>	<b>482,208.37</b>	 FIXED ASSETS	1500-0000		 PROPERTY	1501-0000		ACCUMULATED DEPRECIATION	1600-0000		A/D Buildings	1605-0000	-50,000.00	A/D Exterior	1610-0000	-155.00	A/D Roof	1615-0000	-722.00	A/D Common Area	1620-0000	-145.00	A/D Elevator Upgrades	1625-0000	-158.00	A/D Parking Lots	1635-0000	-122.00	 <b>TOTAL ACCUMULATED DEPRECIATION</b>	<b>1690-0000</b>	<b>-51,302.00</b>	
Account Name	Account Code	Value																																																			
<b>ASSETS</b>	<b>1000-0000</b>																																																				
<b>CURRENT ASSETS</b>	<b>1001-0000</b>																																																				
CASH & CASH EQUIVALENT	1100-0000																																																				
Cash - Operating	1110-0000	482,208.37																																																			
<b>TOTAL CASH &amp; CASH EQUIVALENT</b>	<b>1190-0000</b>	<b>482,208.37</b>																																																			
<b>TOTAL CURRENT ASSETS</b>	<b>1499-0000</b>	<b>482,208.37</b>																																																			
 FIXED ASSETS	1500-0000																																																				
 PROPERTY	1501-0000																																																				
ACCUMULATED DEPRECIATION	1600-0000																																																				
A/D Buildings	1605-0000	-50,000.00																																																			
A/D Exterior	1610-0000	-155.00																																																			
A/D Roof	1615-0000	-722.00																																																			
A/D Common Area	1620-0000	-145.00																																																			
A/D Elevator Upgrades	1625-0000	-158.00																																																			
A/D Parking Lots	1635-0000	-122.00																																																			
 <b>TOTAL ACCUMULATED DEPRECIATION</b>	<b>1690-0000</b>	<b>-51,302.00</b>																																																			

## Account Tree Report with Manipulated Data

This section provides examples of how you can manipulate CFA account tree data in an Excel template for YSR. The example uses a variety of Excel functions to retrieve, sum, and display a custom subset of G/L data: Total Rents and (arbitrarily, for the purpose of illustration) the sum of Total Tenant Rents and Total Operating Expenses - Recoverable.

A	B	C
1 <b>Total Revenue</b>	<b>Sum of Total Rent and Total Recoverables</b>	
2 \$1,444,787.67	\$1,450,521.11	

CF / CFSummary Final Report Design

The example report dumps a sub-set of account data in one Excel worksheet and manipulates it in another. The Excel functions used in this illustration include MATCH, INDEX, ISNA, and IF ISNA. You can adapt these methods, however, to serve other report design purposes.

### To add an account tree report with manipulated data

- 1 Review your CFA report in Voyager.

Custom Financial Reports

Property	comoff01	Asset Manager		Denominator
Book	Accrual	Assistant Manager		Department
Report Template	CFAIS	Client		Source
Period	12/2007	Country		Immediate Sub
Summary By	Client	District Manager		DecimalDigits
Account Tree	is_ysr	Fund Objective		Show Account
<input type="checkbox"/> Freeze Pane				<input type="checkbox"/> Summary
<b>Extracted Data</b>				
4000-0000	INCOME			
4002-0000	REVENUE			
4003-0000	TENANT RENTS			
4100-0000	OFFICE			
4110-0000	Rent - Office	1,215,418		
4125-0000	Rent - Office Parking	45,810		
4190-0000	TOTAL OFFICE	1,261,228		
4899-0000	TOTAL TENANT RENTS	1,261,228		
5100-0000	Recovery - Operating/Common	183,560		
	TOTAL REVENUE	1,444,788		
6010-0000	COMMON RECOVERABLE			
6015-0000	Cleaning/Janitorial	5,325		
6020-0000	Security	4,020		
6025-0000	Landscaping	3,564		
6030-0000	Fire System & Extinguisher	1,125		
6055-0000	Insurance	37,830		
6090-0000	TOTAL COMMON RECOVERABLE	51,864		
6100-0000	REAL ESTATE TAXES - RECOVERABLE			
6110-0000	Real Estate Taxes	89,106		

**2 Review your CFA report design.**

The screenshot shows the 'Custom Financial Reports Layout' window. At the top, there are fields for 'Code' (CFAIS), 'Type' (Account Tree), and 'Notes'. Below these are fields for 'Name' (CFA Income Statement for use in YSR), 'Account Tree' (is\_ysr), 'Report For Invest. Mngt. Module' (unchecked), 'Columns' (1), and 'Apply User Group Security' (unchecked). Below the main configuration area are buttons for 'Edit', 'New', 'Close', 'Help', 'Delete', 'Save As', and 'Set Permissions'. A large section titled 'Financial Formulas' contains a table with columns for 'ID', 'From Acct', 'To Acct', 'Column 1', and 'Delete'. The table has 7 rows, each containing account numbers and descriptions like 'Extracted D...', 'QTD', etc. There are also buttons for 'New Row', 'Function' (Copy Columns), and 'Execute'.

**3 Review the custom account tree used by the CFA report. Make note of which accounts include both a number and a description. In a custom account tree, some accounts may not have account numbers.**

The screenshot shows the 'Account Tree' configuration window. On the left, a tree view shows account categories like '5500-0000: Minority Interest', '5600-0000: Income from Investments', 'RECOVERY INCOME', '5900-0000: TOTAL RECOVERY INCOME', 'TOTAL REVENUE', and 'OPERATING EXPENSES'. On the right, a 'Tree Account (T)' panel shows a 'Number' field (empty) and a 'Description' field (TOTAL REVENUE).

**4 Prepare your YSR report template in Excel. The template will have three worksheets (data, summary, and final report view).**

**a On the first worksheet, add smart markers to retrieve account tree data from the CFA report.**

The screenshot shows an Excel spreadsheet with four columns labeled A, B, C, and D. Row 1 contains the formula '&=[CF].[acctdesc]' in cell A1, '&=[CF].[acctcode]' in cell B1, and '&=[CF].[column1]' in cell C1. The rest of the cells in the first row are empty. The status bar at the bottom shows tabs for 'CF', 'CFSummary', and 'Final Report Design'.

- b** On the second worksheet, use Excel functions to pick out data and manipulate it.

	A	B	C	D
1	<b>Value Column</b>	Is NA	Summable Value	Actual Sum
2	=INDEX(CF!\$C\$1:\$C\$200, MATCH("TOTAL REVENUE",CF!\$B\$1:\$B\$20 0,0))	=ISNA(A2)	=IF(B2=TRUE,0,A2)	=C2
3	MATCH("TOTAL TENANT RENTS",CF!\$B\$1:\$B\$200,0 ))	=ISNA(3)	=IF(B3=TRUE,0,A3)	=C3
4	=INDEX(CF!\$C\$1:\$C\$200, MATCH("TOTAL OPER EXP RECOVERABLE",CF!\$B\$1:\$ B\$200,0))	=ISNA(A4)	=IF(B4=TRUE,0,A4)	=SUM(C3:C4)
5				
6	This column retrieves, from the CF worksheet, the specific account values needed for the report.	In case there are no results in column B, use the ISNA function to return TRUE or FALSE.	If ISNA=TRUE, return 0; otherwise return the value identified in column A.	Use this column to sum values as necessary. Here we sum the total tenant rents and total recoverables only.

CF CFSummary Final Report Design

- c** Create named ranges for the cells that contain the final values for the report so that you can refer to them on the next worksheet. For example:
- Cell D2 is given a named range of **CFTotalRev**.
  - Cell D4 is given a named range of **CFRentRecov**.
- d** On the third worksheet, retrieve the values in the named ranges and format as desired for the final report.

	A	B
1	<b>Total Revenue</b>	<b>Sum of Total Rent and Total Recoverables</b>
2	=CFTotalRev	=CFRentRecov

CF CFSummary Final Report Design

- e** Save your Excel template to the Reports path.

**5 Add a YSR report.**

The dialog box contains the following fields:

- Code:** cfa\_is
- Description:** CFA Inc Statement Manipulation
- Top-Level Select Statement:** (empty)
- OR Script File:** (empty)
- Key Column:** (empty)
- Inactive:**
- Exclude From Generate List:**
- Buttons:** Save, New, Save Copy, Define Filters, Map Top-Level Filter, Dump SQL, Delete Setup, Help

Below the main area is a tab bar with the following tabs: Report Setup, Attachment & Email, Additional Attachment & Email, Output Options, Digital Signature. The Report Setup tab is selected.

Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?
1	cfa_is	cfa_is_template.xlsx	(empty)	property	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**6 Add custom filter fields to your YSR report.**

The dialog box contains the following fields:

- YSR Report:** CFA Inc Statement Manipulation (cfa\_is)
- Buttons:** Save, Preview, Help

Below the main area is a table for defining filters:

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?
Property	Property	1	Lookup List	ysiPropertyLookup	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
FromDate	From Date	2	Post Month		<input type="checkbox"/>		<input type="checkbox"/>
ToDate	To Date	3	Post Month		<input type="checkbox"/>		<input type="checkbox"/>

**7 Select Custom Financials as your data source.**

The dialog box contains the following fields:

- YSR Report:** CFA Balance Sheet (cfabs)
- Report Code:** cfabs
- Buttons:** Save, Close, Help

Below the main area is a tab bar with the following tabs: Report Sections, Attachment Setup. The Report Sections tab is selected.

Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns
CF	Custom Financials	(empty)	Custom Financials	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	(empty)
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

- 8** Map filters. Map the **Template** element in the **Standard Analytics Report Filters** column to the code of the CFA report you want to generate in YSR. Enter the CFA report code in the **Constant Value** column.

Map Standard Analytics Filter

YSR Report:	CFA Inc Statement Manipulation (cfais)	
Report Code:	cfais	
SectionCode:	CF	
<input type="button" value="Save"/> <input type="button" value="Close"/> <input type="button" value="Help"/>		
 <input type="button" value="Print"/>		
Standard Analytics Report Filters	YSR Report Filters	Constant Value
Template	Property	CFAIS
PropertyCode	FromDate	
FromMMYY	ToDate	
ToMMYY		



If your YSR report has a top-level **Book** filter field (supported by `ysiBookLookup`), map the **Book** filter field to the **BookCodeList** element in the **Standard Analytics Report Filters** column. If you fail to specify a book for the **BookListCode** element, CFA defaults to the Cash book.

- 9** Generate your YSR report.

A	B	C
1 Total Revenue	Sum of Total Rent and Total Recoverables	
2 \$1,444,787.67	\$1,450,521.11	

◀ ▶ ⏪ ⏩ CF / CFSummary Final Report Design ⚙

## CFA Property Portfolio Reports in YSR

### In this section:

Property Portfolio Report, Summarized by Attribute .....	182
Property Portfolio Report, Summarized by Property.....	186

Property Portfolio reports help you aggregate and analyze data for multiple properties at a time. You can analyze data by summarizing data by attribute (by Region, for example) or by breaking data out per property.

In Voyager CFA, both types of analysis are supported with the same report type and same report filter. With YSR, however, the two types of analysis require separate report designs. This is because YSR does not support the dual purposes of the **Summary By** filter component of the CFA Property Portfolio report filter, which includes both attributes and **Property or Entity**, as illustrated in the following graphic:

Rent	Vacancy	Rent Less Vac	EO%
230,450	1,780	228,670	99
<b>230,450</b>	<b>1,780</b>	<b>228,670</b>	<b>99</b>
189,650	31,000	158,650	84
<b>189,650</b>	<b>31,000</b>	<b>158,650</b>	<b>84</b>
420,100	32,780	387,320	92

With YSR, you can build a filter field that displays both attribute values and **Property or Entity**, but you cannot map the list values correctly. Therefore you must create different report designs in YSR depending on whether you want to summarize data by attribute or break data out by property.

Property Portfolio reports display data in a fixed number of columns, as determined by the CFA report design. (Comparison reports, by contrast, display data in multiple columns depending on user input.) Because report data appears in just one data column, report design for Property Portfolio reports is relatively simple. No additional scripting is necessary.

## Property portfolio reports and account code masking

All CFA reports return data for the field names **AcctCode** and **AcctDesc**. For all report types other than Property Portfolio reports, these field names return GL account numbers and GL account descriptions. Because the Property Portfolio reports summarize data differently, the **AcctCode** and **AcctDesc** fields used in Portfolio reports return attribute values or property codes plus descriptions. In this way, attribute values or property codes (rather than GL accounts) appear as row headers.

Furthermore, **AcctCode** data appears with GL account format masking (string-string). For example, the property code comoff01 appears as como-ff01. To display **AcctCode** data correctly, use Excel's SUBSTITUTE function to remove the hyphen.

Alternatively, use **AcctDesc** data only.

## Property Portfolio Report, Summarized by Attribute

This section describes how to create a property portfolio report, summarized by attribute, for use in YSR.

The example report is based on a simple CFA report design, illustrated here:

The screenshot shows the 'Custom Financial Reports Layout' window. At the top, it displays the report type as 'Property Portfolio'. Below this, there are fields for 'Name' (Economic Occupancy), 'Account Tree' (ysi\_is), and 'Columns' (4). There are also checkboxes for 'Report For Invest. Mgmt. Module' and 'Apply User Group Security'. At the bottom of the window are buttons for 'Edit', 'New', 'Close', 'Help', 'Delete', 'Save As', 'Set Permissions', and 'Dump SQL'.

Below the main window is a 'Financial Formulas' section. It includes a toolbar with 'New Row', 'Function', 'Copy Columns', 'From Column 1', 'to Column 1', and 'Execute'. A table below the toolbar contains four columns labeled 'Column 1', 'Column 2', 'Column 3', and 'Column 4'. The table has two rows of data:

ID	From Acct	To Acct	Column 1	Column 2	Column 3	Column 4	Delete
1			Total Rent	Vacancy	Rent Less V...	EO%	
2			MTD	MTD	Formula	Formula	

The example YSR report uses an Excel report template, designed as follows:

The screenshot shows an Excel spreadsheet with a table structure. The table has columns labeled A through E. Row 1 contains column headers. Row 2 contains the text 'Economic Occupancy by Market'. Row 4 contains the header 'Summarize By'. Row 5 contains the formula '&=[CF].[acctDesc]'. Row 6 contains the formula '&=[CF].[column1]'. Row 7 contains the formula '&=[CF].[column2]'. Row 8 contains the formula '&=[CF].[column3]'. Row 9 contains the formula '&=[CF].[column4]'. The table structure is as follows:

A	B	C	D	E	
1					
2	Economic Occupancy by Market				
3					
4	Summarize By	Total Rent	Vacancy	Rent Less Vac	EO %
5	&=[CF].[acctDesc]	&=[CF].[column1]	&=[CF].[column2]	&=[CF].[column3]	&=[CF].[column4]
6					

**To set up a property portfolio report, summarized by attribute**

- 1 Review your CFA report in Voyager.

Custom Financial Reports					
Property	comoff01^comoff	Asset Manager		Denominator	
Book	accrual	Assistant Manager		Department	
Report Template	EO	CFDA		Source	
Period	02/2008 to 01/2017	Client		Immediate Source	
Summary By	Market	Country		DecimalDigits	0
Account Tree	ysi_is	District Manager		Show Code	Hide
				Summary	<input type="checkbox"/> Tree Level
				Freeze Pane	<input type="checkbox"/>
		Total Rent	Vacancy	Rent Less Vac	EO%
<b>Raleigh/Durham</b>		230,450	1,780	228,670	99
Roosevelt Tower		230,450	1,780	228,670	99
<b>Total Raleigh/Durham</b>					
<b>Phoenix</b>		189,650	31,000	158,650	84
Sunrise Tower		189,650	31,000	158,650	84
<b>Total Phoenix</b>					
<b>Grand Total</b>		420,100	32,780	387,320	92

- 2 Add a YSR report.

YSR Report Setup																																				
Code	cfa_eo_by_market			Notes																																
Description	CFA Economic Occupancy by Market Attribute																																			
Top-Level Select Statement																																				
Key Column																																				
Inactive	<input type="checkbox"/>	Exclude From Generate List <input type="checkbox"/>																																		
<input type="button" value="Save"/> <input type="button" value="New"/> <input type="button" value="Save Copy"/> <input type="button" value="Define Filters"/> <input type="button" value="Map Top-Level Filter"/> <input type="button" value="Dump SQL"/> <input type="button" value="Delete Setup"/> <input type="button" value="Help"/>																																				
<input type="button" value="Report Setup"/> <input type="button" value="Attachment &amp; Email"/> <input type="button" value="Additional Attachment &amp; Email"/> <input type="button" value="Output Options"/> <input type="button" value="Digital Signature"/>																																				
<table border="1"> <thead> <tr> <th>Order</th> <th>Report Code</th> <th>Template File</th> <th>Script File</th> <th>Page Break Column</th> <th>Map Script Filter</th> <th>Sections</th> <th>Inactive?</th> <th>Delete?</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>eo</td> <td>ysr_cfa_eobymarket.xlsx</td> <td></td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </tbody> </table>										Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?	1	eo	ysr_cfa_eobymarket.xlsx					<input type="checkbox"/>	<input type="checkbox"/>								<input type="checkbox"/>	<input type="checkbox"/>
Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?																												
1	eo	ysr_cfa_eobymarket.xlsx					<input type="checkbox"/>	<input type="checkbox"/>																												
							<input type="checkbox"/>	<input type="checkbox"/>																												

### 3 Define the top-level filter fields for your YSR report.

**Report Filters Setup**

Merged Report: CFA Economic Occupancy by Market Attribute (cfa\_eo\_by\_market)

Save Preview Help

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	List Values	Code to ID
Property	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		
CFType	CF Type	2	List		<input checked="" type="checkbox"/>	Portfolio by Attribute	
PostMonth	Post Month	3	Post Month		<input checked="" type="checkbox"/>		
Book	Book	4	Lookup List	ysiBookLookup	<input checked="" type="checkbox"/>		
SuppressZero	Suppress Zero	5	Checkbox		<input checked="" type="checkbox"/>		
AttributeSortName	Summarize by	6	Lookup List		<input checked="" type="checkbox"/>	Select A.sname CodeField, Select A.hmy IDF	
Detail	Detail	7	List		<input checked="" type="checkbox"/>	0^1	
					<input checked="" type="checkbox"/>		
					<input checked="" type="checkbox"/>		

**CFType** Optional. Used here as a display field. By including the CFA type here, you can retrieve the associated list value (Portfolio by Attribute) with another mini script. This can be useful, for example, if you want to display the CFA report type in your report header. If you add this field and refer to it in a script, you must map it (on the filter mapping screen associated with the script).

**AttributeSortName** Required. You must add a filter field that supplies Voyager with an attribute for summarizing data.

There are multiple ways to supply attribute names. In this example, a lookup list is defined as follows:

**List Values** SELECT A.sname CodeField, A.sName TextField  
FROM AttributeName A WHERE 1=1

**Code to ID** SELECT A.Hmy IDFField FROM AttributeName A  
WHERE 1=1 and A.sname IN ([CodeField])

### 4 On the **Report Sections Setup** screen, select Custom Financials as your data source.

**Report Sections Setup**

YSR Report: CFA Economic Occupancy by Market Attribute (cfa\_eo\_by\_market)  
Report Code: eo

Save Close Help

Report Sections		Attachment Setup					
Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns
CF	CF		Custom Financials	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

## 5 Map filters.

**Map Standard Analytics Filter**

YSR Report: CFA Economic Occupancy by Market Attribute (cfa\_eo\_by\_market)  
 Report Code: eo  
 SectionCode: CF

**Standard Analytics Report Filters**

Standard Analytics Report Filters	YSR Report Filters	Constant Value
Template		eo
PropertyCode	Property	
FromMMYY	PostMonth	
ToMMYY	PostMonth	
AttributeSortName	AttributeSortName	
AttributeSortValue	AttributeSortName	
SuppressZero	SuppressZero	
BookCodeList	Book	
Detail	Detail	

**Buttons:** Save, Close, Help



Map the **Book** filter field to the **BookCodeList** element in the **Standard Analytics Report Filters** column. If you fail to specify a book for the **BookListCode** element, CFA defaults to the Cash book.

## 6 Generate your YSR report.

**CFA Economic Occupancy by Market Attribute**

Property	comoff01^comoff02	Report Name	CFA Economic Occupancy by Market Attribute (	▼
CF Type	Portfolio by Attribute	Output Type	Screen	<input type="checkbox"/> Attach Reports
Post Month	01/2017	Merge Reports	<input type="checkbox"/>	<input type="checkbox"/> Email Reports
Book		Show Grid	<input type="checkbox"/>	<input type="checkbox"/> Publish To SharePoint
Suppress Zero	<input checked="" type="checkbox"/>			<input type="checkbox"/> Show on Portal
Summarize by	Market			
Detail	0	<b>Generate</b>	<b>Clear</b>	

**Economic Occupancy by Market**

Summarize By	Total Rent	Vacancy	Rent Less Vac	EO %
<b>Market</b>				
Raleigh/Durham	230450	1780	228670	99.22759818
Phoenix	189650	31000	158650	83.65409966
<b>Total Market</b>	420100	32780	387320	92.19709593
<b>Grand Total</b>	420100	32780	387320	92.19709593

## Property Portfolio Report, Summarized by Property

Property Portfolio reports summarized by property have nearly the same setup requirements as Property Portfolio reports summarize by attributes (illustrated in the previous section).

The two types of report can use the same CFA report template, the same Excel report template, and the same YSR report sections. They differ only with respect to filtration. Filtration for the property-based report is somewhat less complicated because no attribute filter is required.

This section shows the top-level filter field definition and filter mapping screens for a Property Portfolio report summarized by property. All other setup requirements of this report are the same as for the Property Portfolio report, summarized by attribute.

### Filter Definition

**Report Filters Setup**

Merged Report: CFA Economic Occupancy by Property (cfa\_eo\_by\_prop)

**Save** **Preview** **Help**

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	List Values
Property	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
CFType	CFType	2	List		<input type="checkbox"/>		<input type="checkbox"/>	Portfolio by Property
PostMonth	Post Month	3	Post Month		<input type="checkbox"/>		<input type="checkbox"/>	
Book	Book	4	Lookup List	ysiBookLookup	<input type="checkbox"/>		<input type="checkbox"/>	
SuppressZero	Suppress Zero	5	Checkbox		<input type="checkbox"/>		<input type="checkbox"/>	

### Filter Mapping

**Map Standard Analytics Filter**

YSR Report: CFA Economic Occupancy by Property (cfa\_eo\_by\_prop)

Report Code: 1

SectionCode: CF

**Save** **Close** **Help**

Standard Analytics Report Filters	YSR Report Filters	Constant Value
Template	Property	eo
PropertyCode	PostMonth	
FromMMYY	PostMonth	
ToMMYY	PostMonth	
BookCodeList	Book	
Detail		1
SuppressZero	SuppressZero	

## CFA Comparison Reports in YSR

### In this section:

Property Comparison Reports in YSR .....	187
Attribute Comparison Reports in YSR .....	194

CFA Comparison Reports provide side-by-side comparisons of data in multiple columns, as determined by the user at run time. For example, the following graphic shows a CFA Attribute Comparison report with two columns in its report design (MTD and YTD data). When the user generates the report, Voyager dynamically generates one column per property, plus a Total column per data type (Total MTD/ Total YTD).

The screenshot shows the 'Custom Financial Reports' dialog box. On the left, there are several configuration fields: 'Property' (set to 'comoff01^comoff'), 'Book' (set to 'accrual'), 'Report Template' (set to 'GLAttCo'), 'Period' (set to '02/2008 to 01/2017'), 'Comparison By' (set to 'Market'), and 'Account Tree' (set to 'ysi\_bs'). To the right of these are dropdown menus for 'Asset Manager', 'Assistant Manager', 'CFDA', 'Client', 'Country', and 'District Manager'. Further right are settings for 'Denominator', 'Department', 'Source', and 'Immediate Source'. Below these are controls for 'DecimalDigits' (set to 0), 'Suppress Zero' (checked), 'Show Account' (set to 'Hide'), 'Detail' (checked), and 'Summary' (unchecked). There are also buttons for 'Tree Level' (set to 1) and 'Grid' (unchecked). At the bottom of the dialog is a preview pane showing a table structure with columns for Raleigh/Durham MTD, Raleigh/Durham YTD, Phoenix MTD, Phoenix YTD, Total MTD, and Total YTD. The table has sections for ASSETS, CURRENT ASSETS, CASH & CASH EQUIVALENT, and various totals.

	Raleigh/Durham MTD	Raleigh/Durham YTD	Phoenix MTD	Phoenix YTD	Total MTD	Total YTD
<b>ASSETS</b>						
<b>CURRENT ASSETS</b>						
CASH & CASH EQUIVALENT						
Cash - Operating	228,670	228,670	158,650	158,650	387,320	387,320
<b>TOTAL CASH &amp; CASH EQUIVALENT</b>	<b>228,670</b>	<b>228,670</b>	<b>158,650</b>	<b>158,650</b>	<b>387,320</b>	<b>387,320</b>
<b>TOTAL CURRENT ASSETS</b>	<b>228,670</b>	<b>228,670</b>	<b>158,650</b>	<b>158,650</b>	<b>387,320</b>	<b>387,320</b>
<b>TOTAL ASSETS</b>	<b>228,670</b>	<b>228,670</b>	<b>158,650</b>	<b>158,650</b>	<b>387,320</b>	<b>387,320</b>
<b>LIABILITIES &amp; EQUITY</b>						

Because the number of columns in comparison reports is not known ahead of time, comparison reports in YSR require additional scripting to generate column headers. Example scripts are provided, with commentary, in the following sections.

## Property Comparison Reports in YSR

The example report provided in this section details a CFA report that compares MTD and YTD data. The CFA report design has just two columns (MTD and YTD), but the report columns expand at run time to include two columns per property (MTD and YTD) plus two Total columns (Total MTD and Total YTD).

### Property Comparison Column Headings Script

To accommodate the dynamic expansion of columns at run time, this report uses the following script to generate column headings:

```

//SELECT PropHead
SELECT
  (RTRIM(LTRIM(P.sCode)) + CHAR(13) + CHAR(10) + CFHead.sCalcFormula)
AS ColumnLabel,
  (p.sCode) AS OrderingStr
FROM (
  SELECT sCode
  FROM PROPERTY
  WHERE 1 = 1 and Property.iType =3
  #Condition1#
) as P,

(SELECT
  GLTemplateCell.iCol,
  GLTemplateCell.sCalcFormula
FROM GLTemplate
INNER JOIN GLTemplateCell
  ON GLTemplate.hMy = GLTemplateCell.hTemplate
WHERE (GLTemplate.sCode = '#Template#')
AND (GLTemplateCell.iRow = 1)) AS CFHead

UNION ALL

SELECT
  ('Total' + CHAR(13) + CHAR(10) + CFHead.sCalcFormula) AS ColumnLabel,
  'zzzzzzzzzz' AS OrderingStr
FROM (
  SELECT
    GLTemplateCell.iCol,
    GLTemplateCell.sCalcFormula
  FROM GLTemplate
  INNER JOIN GLTemplateCell
    ON GLTemplate.hMy = GLTemplateCell.hTemplate
  WHERE (GLTemplate.sCode = '#Template#')
  AND (GLTemplateCell.iRow = 1)
) AS CFHead

ORDER BY 2
//END SELECT

```

This script retrieves column headers in the order required by the CFA data source (by property scode, followed by 'Total'). For illustration, if a user generates the report for properties comind01^comoff01, the script produces these column headings:

	ColumnLabel	OrderingStr
1	comind01 YTD	comind01
2	comind01 MTD	comind01
3	comoff01 MTD	comoff01
4	comoff01 YTD	comoff01
5	Total MTD	zzzzzzzzzz
6	Total YTD	zzzzzzzzzz

For users who are unfamiliar with SQL, some explanatory points follow:

- The script retrieves the column headings associated with the CFA template (MTD, YTD) and the property scodes corresponding to the user's run time selections.
- It also retrieves the string 'Total' for as many instances as there are column headings in the CFA template, using a UNION ALL clause.
- The script concatenates the column headings (MTD, YTD) with property scodes and 'Total' and aliases the resulting strings (prop1 MTD, prop2 YTD, Total MTD, Total YTD) as ColumnLabel.
- Finally, the script uses property scodes as an ordering mechanism. (Where the script retrieves 'Total,' it supplies 'zzzzzzzz' as the ordering value so that the Total labels appear last in the sort order.)
- The result is a list of column headings, aliased ColumnLabel, ordered by property scodes, followed by Total headings. To make use of the column headings, add a ColumnLabel smart marker somewhere in your Excel template. (At run time, the smart marker expands to display all the column labels required for the report.) Then, in the cells where you want the labels to appear, use an Excel formula to refer to the corresponding column label.

### To set up a CFA Property Comparison report in YSR

#### 1 Review your CFA report in Voyager.

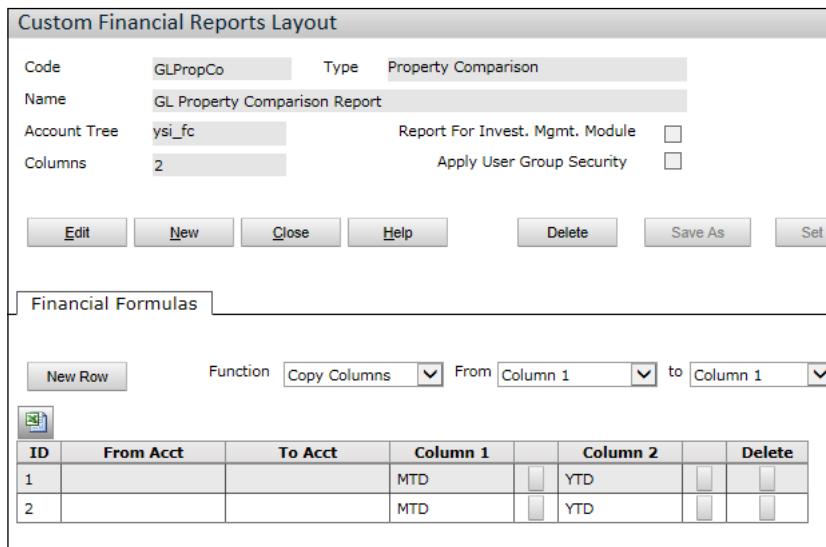
The screenshot shows the 'Custom Financial Reports' dialog in Voyager. The configuration includes:

- Property:** comoff01^comoff
- Book:** accrual
- Report Template:** GLPropCo
- Period:** 02/2008 to 01/2017
- Summary By:** Asset Manager
- Account Tree:** ysi\_fc
- Denominator:** (empty)
- Department:** (empty)
- Source:** (empty)
- Immediate Source:** (empty)
- DecimalDigits:** 0
- Suppress:** (checkbox)
- Show Account:** Hide
- Hide:** (checkbox)
- Summary:** (checkbox)
- Tree Level:** 1

The main preview area displays a table with the following data:

	comoff01 MTD	comoff01 YTD	comoff02 MTD	comoff02 YTD	Total MTD	Total YTD
Cash - Operating	158,650	158,650	228,670	228,670	387,320	387,320
Gross Potential Rent	189,650	189,650	230,450	230,450	420,100	420,100
Less: Vacancies	-31,000	-31,000	-1,780	-1,780	-32,780	-32,780
Reserve Acct	0	0	0	0	0	0

- 2 Review your CFA report design. Take note of the report code (glpropco in this example), as you must hard-code the report code into your YSR report design.



- 3 Prepare a YSR report template in Excel with two worksheets.

- a On one worksheet, add smart markers to retrieve column headings from the custom script included in this example.

	A	B
1	Labels	
2	&=[ColHead].[ColumnLabel]	
3		
4		

- b On the other worksheet, design the body of the report. Add a sufficient number of columns to accommodate the CFA report design and the user's runtime filter criteria. For example, a CFA report with two data columns, when generated for a property list containing 4 properties, requires 10 columns (4 × 2 data columns + 2 Total columns).

	A	B	C	D	
1	Property Comparison Report				
2					
3					
4	Acct Code	Acct Desc	&=[ColHead].[ColumnLabel]	=IF(Sheet2!A3="","",Sheet2!A3)	=IF(Sheet2!A3="","",Sheet2!A3)
5	&=[CF].[acctcode]	&=[CF].[acctDesc]	&=[CF].[column1]	&=[CF].[column2]	&=[CF].[column3]
6					

Cell C4

Uses an Excel reference to retrieve the first column label from Sheet2.

=Sheet2!A2

**Cell D4** Uses conditional logic to check whether there is a column label in Sheet2, cell A3. If there is, Excel displays the label here. If not, the cell remains blank.

=IF(Sheet2!A3="","",Sheet2!A3)

**TIP** If you do not use conditional logic, but simply enter a reference to a cell that (potentially, depending on the user's runtime selection) contains a column label, **0** appears as the column heading for blank columns.

- 4 Add a YSR report. The script file in this example (CFheader.txt) contains the SELECT statement for retrieving column headers (illustrated at the beginning of this section).

The screenshot shows the 'YSR Report Setup' dialog box. In the top-left, the 'Code' field is set to 'cfa\_glpco'. The 'Description' field contains 'CFA GL Property Comparison report for YSR'. Below these are fields for 'Top-Level Select Statement' and 'OR Script File'. To the right is a 'Notes' text area. At the bottom are buttons for 'Save', 'New', 'Save Copy', 'Define Filters', 'Map Top-Level Filter', 'Dump SQL', 'Delete Setup', and 'Help'.

Below the dialog is a table titled 'Report Setup' with the following data:

Report Setup									
	Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?
	1	glpc	GLPropComp.xlsx	CFheader.txt				<input type="checkbox"/>	<input type="checkbox"/>
								<input type="checkbox"/>	<input type="checkbox"/>

**5 Define the top-level report filter fields. (Define Filters button)**

Report Filters Setup

Merged Report: CFA GL Property Comparison report for YSR (cfa\_glpco)

Save Preview Help

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	List Values
Property	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
CFType	CF Type	2	List		<input type="checkbox"/>		<input type="checkbox"/>	Prop Comparison
PostMonth	Post Month	3	Post Month		<input type="checkbox"/>		<input type="checkbox"/>	
Book	Book	4	Lookup List	ysiBookLookup	<input type="checkbox"/>		<input type="checkbox"/>	
SuppressZero	Suppress Zero	5	Checkbox		<input type="checkbox"/>		<input type="checkbox"/>	
Template	Template	6	List		<input type="checkbox"/>		<input type="checkbox"/>	GLPropCo
Detail	Detail	7	List		<input type="checkbox"/>		<input type="checkbox"/>	0^1
					<input type="checkbox"/>		<input type="checkbox"/>	
					<input type="checkbox"/>		<input type="checkbox"/>	
					<input type="checkbox"/>		<input type="checkbox"/>	

- CFType** Optional. Used here as a display field. By including the CFA type here, you can retrieve the associated list value (Prop Comparison) with another mini script. This can be useful, for example, if you want to display the CFA report type in your report header. If you add this field and refer to it in a script, you must map it (on the filter mapping screen associated with the script).
- Template** Included here not for retrieving the user's selection, but so that the column headers script can refer to the template scode (GLPropCo, defined in the List Values column).
- Detail** Optional. Mirrors the design of the CFA report. In a later step, this filter field is mapped to the Detail filter element of the CFA data source.

**6 Add report sections. Property Comparison reports require at minimum one section for retrieving data from Custom Financials and one scripted section for retrieving column headers.**

Report Sections Setup

YSR Report: CFA GL Property Comparison report for YSR (cfa\_glpco)  
Report Code: glpc

Save Close Help

Report Sections		Attachment Setup						
Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	Primary?
CF	CF		Custom Financials	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
ColHead	Column Headers	PropHead		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**7** Map filters for the Custom Financials data source.

**Map Standard Analytics Filter**

YSR Report:	CFA GL Property Comparison report for YSR (cfa_glpropco)	
Report Code:	glpc	
SectionCode:	CF	
<input type="button" value="Save"/> <input type="button" value="Close"/> <input type="button" value="Help"/>		
 <input type="button" value="Print"/>		
Standard Analytics Report Filters	YSR Report Filters	Constant Value
Template	Property	GLPropCo
PropertyCode	PostMonth	
ToMMYY	PostMonth	
FromMMYY	Book	
BookCodeList	SuppressZero	
SuppressZero		
TreeCode		ysi_fc
Detail	Detail	



Map the **Book** filter field to the **BookCodeList** element in the **Standard Analytics Report Filters** column. If you fail to specify a book for the **BookListCode** element, CFA defaults to the Cash book.

**8** Map filters for the mini script (return to the **YSR Report Setup** screen).

**Map Script Filter**

YSR Report:	CFA GL Property Comparison report for YSR (cfa_glpropco)	
Report Code:	glpc	
<input type="button" value="Save"/> <input type="button" value="Close"/> <input type="button" value="Help"/>		
 <input type="button" value="Print"/>		
YSR Report Filters	Script Filter Condition	
Property (#Condition1#)	Property.Hmy in (#Property#)	
Template (#Condition6#)	#Template#	

**Property (#Condition1#)** Required. Establishes a mapping between the top-level Property filter field and the script that retrieves column headers.

**Template (#Condition6#)** Required in this example report because the script that retrieves data for column headings uses a token to refer to the CFA template code. By using a token, the script remains flexible and can be adopted for other CFA-based reports.

Alternatively, you can hard-code a CFA template code into the script you use for retrieving column headings. The template code is necessary in the script in order to retrieve the column headings (e.g., MTD, YTD) appropriate to the CFA report template.

## 9 Generate your YSR report.

**CFA GL Property Comparison report for YSR**

Property	comoff01^comoff02	Report Name	CFA GL Property Comparison report for YSR (cf)																																
CF Type	Prop Comparison	Output Type	Screen																																
Post Month	01/2017	Merge Reports	<input type="checkbox"/>																																
Book	accrual	Show Grid	<input type="checkbox"/>																																
Suppress Zero	<input checked="" type="checkbox"/>	Attach Reports	<input type="checkbox"/>																																
Template	GLPropCo	Email Reports	<input type="checkbox"/>																																
Detail	0	Publish To SharePoint	<input type="checkbox"/>																																
<input type="button" value="Generate"/> <input type="button" value="Clear"/>																																			
<input type="button" value="Sheet1"/> <input type="button" value="Sheet2"/>																																			
<b>Property Comparison Report</b> <table border="1"> <thead> <tr> <th>Acct Code</th> <th>Acct Desc</th> <th>comoff01 YTD</th> <th>comoff01 MTD</th> <th>comoff02 MTD</th> <th>comoff02 YTD</th> <th>Total MTD</th> <th>Total YTD</th> </tr> </thead> <tbody> <tr> <td>1110-0000</td> <td>Cash - Operating</td> <td>\$ 158,650.00</td> <td>\$ 158,650.00</td> <td>\$ 228,670.00</td> <td>\$ 228,670.00</td> <td>\$ 387,320.00</td> <td>\$ 387,320.00</td> </tr> <tr> <td>4410-0000</td> <td>Gross Potential Rent</td> <td>\$ 189,650.00</td> <td>\$ 189,650.00</td> <td>\$ 230,450.00</td> <td>\$ 230,450.00</td> <td>\$ 420,100.00</td> <td>\$ 420,100.00</td> </tr> <tr> <td>4450-0000</td> <td>Less: Vacancies</td> <td>\$ (31,000.00)</td> <td>\$ (31,000.00)</td> <td>\$ (1,780.00)</td> <td>\$ (1,780.00)</td> <td>\$ (32,780.00)</td> <td>\$ (32,780.00)</td> </tr> </tbody> </table>				Acct Code	Acct Desc	comoff01 YTD	comoff01 MTD	comoff02 MTD	comoff02 YTD	Total MTD	Total YTD	1110-0000	Cash - Operating	\$ 158,650.00	\$ 158,650.00	\$ 228,670.00	\$ 228,670.00	\$ 387,320.00	\$ 387,320.00	4410-0000	Gross Potential Rent	\$ 189,650.00	\$ 189,650.00	\$ 230,450.00	\$ 230,450.00	\$ 420,100.00	\$ 420,100.00	4450-0000	Less: Vacancies	\$ (31,000.00)	\$ (31,000.00)	\$ (1,780.00)	\$ (1,780.00)	\$ (32,780.00)	\$ (32,780.00)
Acct Code	Acct Desc	comoff01 YTD	comoff01 MTD	comoff02 MTD	comoff02 YTD	Total MTD	Total YTD																												
1110-0000	Cash - Operating	\$ 158,650.00	\$ 158,650.00	\$ 228,670.00	\$ 228,670.00	\$ 387,320.00	\$ 387,320.00																												
4410-0000	Gross Potential Rent	\$ 189,650.00	\$ 189,650.00	\$ 230,450.00	\$ 230,450.00	\$ 420,100.00	\$ 420,100.00																												
4450-0000	Less: Vacancies	\$ (31,000.00)	\$ (31,000.00)	\$ (1,780.00)	\$ (1,780.00)	\$ (32,780.00)	\$ (32,780.00)																												

## Attribute Comparison Reports in YSR

The example report provided in this section details a CFA report that compares MTD and YTD data. The CFA report design has just two columns (MTD and YTD), but the report columns expands at run time to include two columns per property (MTD and YTD) plus two Total columns (Total MTD and Total YTD).



This example report does not support filtration by attribute only. The user must filter by a property or property list to return data.

### Attribute Comparison Column Headings Script

To accommodate the dynamic expansion of columns at run time, this report uses the following script to generate column headings:

```
//SELECT AttrHead
SELECT
    (AVSub.sValue + CHAR(13) + CHAR(10) + CFHead.sCalcFormula) AS Column-
Label,
    (AVSub.hMy * 1000 + CFHead.iCol) AS OrderingNum
FROM (
    SELECT DISTINCT
        AV.sValue,
        AV.hMy
    FROM AttributeValue AS AV
    INNER JOIN AttributeName AS AN
    ON AV.hAttributeName = AN.hMy
    INNER JOIN AttributeXref
    ON AV.hMy = AttributeXref.hAttributeValue
    INNER JOIN PROPERTY
    ON AttributeXref.hFileRecord = Property.HMY
```

```

WHERE (AN.iFileType = 3)
#Condition7#
#Condition1#
) AS AVSUB,

(SELECT
GLTemplateCell.iCol,
GLTemplateCell.sCalcFormula
FROM GLTemplate
INNER JOIN GLTemplateCell
ON GLTemplate.hMy = GLTemplateCell.hTemplate
WHERE (GLTemplate.sCode = '#Template#')
AND (GLTemplateCell.iRow = 1)) AS CFHead

UNION ALL

SELECT
('Total' + CHAR(13) + CHAR(10) + CFHead.sCalcFormula) AS ColumnLabel,
(1000000001 + CFHead.iCol) AS OrderingNum
FROM (
SELECT
GLTemplateCell.iCol,
GLTemplateCell.sCalcFormula
FROM GLTemplate
INNER JOIN GLTemplateCell
ON GLTemplate.hMy = GLTemplateCell.hTemplate
WHERE (GLTemplate.sCode = '#Template#')
AND (GLTemplateCell.iRow = 1)
) AS CFHead

ORDER BY 2
//END SELECT

```

This script retrieves column headers in the order required by the CFA data source (by the hm of the attribute value, followed by 'Total'). For illustration, if a user generates the report for properties comoff01^comoff02, the script produces these column headings:

	ColumnLabel	OrderingNum
1	Raleigh/Durham MTD	54001
2	Raleigh/Durham YTD	54002
3	Phoenix MTD	94001
4	Phoenix YTD	94002
5	Total MTD	1000000002
6	Total YTD	1000000003

For users who are unfamiliar with SQL, some explanatory points follow:

- The script retrieves the column headings associated with the CFA template (MTD, YTD) and the attribute values corresponding to the user's run time selections.
- It also retrieves the string 'Total' for as many instances as there are column headings in the CFA template, using a UNION ALL clause.
- The script concatenates column headings (MTD, YTD) and attribute values and aliases the resulting strings (Raleigh/Durham MTD, Raleigh/Durham YTD, Total MTD, Total YTD, etc.) as ColumnLabel.
- Finally, the script uses the attribute value hm as an ordering mechanism. (Where the script retrieves 'Total,' it supplies '1000000001 + Column #' of the CFA report design as the ordering value so that the Total labels appear last and in the correct sort order.)
- The result is a list of column headings, aliased ColumnLabel, ordered by attribute value hm, followed by Total headings. To make use of the column headings, add a ColumnLabel smart marker somewhere in your Excel template. Then, in the cells where you want the labels to appear, use an Excel formula to refer to the corresponding column label.

## To set up a CFA Attribute Comparison report in YSR

- Review your CFA report in Voyager.

The screenshot shows the 'Custom Financial Reports' dialog box. In the 'Property' section, 'Book' is set to 'accrual' and 'Report Template' is 'GLAttCo'. The 'Period' is from '02/2008' to '01/2017'. 'Comparison By' is 'Market'. The 'Account Tree' is 'ysi\_bs'. On the right, there are various filters like 'Asset Manager', 'Denominator', 'Department', etc., with 'DecimalDigits' set to 0 and 'Suppress Zero' checked. Below the filters is a summary table with columns for Raleigh/Durham MTD, Raleigh/Durham YTD, Phoenix MTD, Phoenix YTD, Total MTD, and Total YTD. The main area displays the 'ASSETS' section with a table of financial data:

	Raleigh/Durham MTD	Raleigh/Durham YTD	Phoenix MTD	Phoenix YTD	Total MTD	Total YTD
CURRENT ASSETS						
CASH & CASH EQUIVALENT						
Cash - Operating	228,670	228,670	158,650	158,650	387,320	387,320
TOTAL CASH & CASH EQUIVALENT	228,670	228,670	158,650	158,650	387,320	387,320
TOTAL CURRENT ASSETS	228,670	228,670	158,650	158,650	387,320	387,320
TOTAL ASSETS	228,670	228,670	158,650	158,650	387,320	387,320

- Review your CFA report design. Take note of the report code (glpropco in this example), as you must hard-code the report code into your YSR report design.

The screenshot shows the 'Custom Financial Reports Layout' dialog box. It includes fields for 'Code' (GLAttCo), 'Name' (GL Attribute Comparison), 'Account Tree' (ysi\_bs), 'Type' (Attribute Comparison), 'Columns' (2), and checkboxes for 'Report For Invest. Mgmt. Module' and 'Apply User Group Security'. Below the dialog is a 'Financial Formulas' section with a table for defining formulas:

New Row	Function	Copy Columns	From	Column 1	to	Column 1
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						

- Prepare a YSR report template in Excel with two worksheets.

- On one worksheet, add smart markers to retrieve column headings from the custom script included in this example.

The screenshot shows an Excel spreadsheet with two sheets: 'Sheet1' and 'Sheet2'. The 'Sheet2' tab is active and contains the following custom script:

```

A
1 Labels
2 &=[ColHead].[ColumnName]
3
4

```

- b** On the other worksheet, design the body of the report. Add a sufficient number of columns to accommodate the CFA report design and the user's run time filter criteria. For example, a CFA report with two data columns, when generated for a property list containing 4 properties, requires 10 columns ( $4 \times 2$  data columns + 2 Total columns).

	A	B	C	D
1	<b>Property Comparison Report</b>			
2				
3				
4	<b>Acct Code</b>	<b>Acct Desc</b>	<code>&amp;=[ColHead].[ColumnLabel]</code>	<code>=IF(Sheet2!A3="", "", Sheet2!A3)</code>
5	<code>&amp;=[CF].[acctcode]</code>	<code>&amp;=[CF].[acctDesc]</code>	<code>&amp;=[CF].[column1]</code>	<code>&amp;=[CF].[column2]</code>
6				

Sheet1   Sheet2   (+)

**Cell C3**

Retrieves the first column label from Sheet2.

**Cell D3**

Uses conditional logic to check whether there is a column label in Sheet2, cell A3. If there is, Excel displays the label here. If not, the cell remains blank.

**TIP** If you do not use conditional logic, but simply enter a reference to a cell that (potentially, depending on the user's runtime selection) contains a column label, **0** appears as the column heading for blank columns.

- 4** Add a YSR report. The script file in this example (CFheader.txt) contains the SELECT statement for retrieving column headers (illustrated at the beginning of this section).

**YSR Report Setup**

Code	cfa_glatco	x	Notes																																
Description	CFA GL Attribute Comparison report for YSR																																		
Top-Level Select Statement																																			
OR Script File																																			
Key Column																																			
Inactive	<input type="checkbox"/>	Exclude From Generate List	<input type="checkbox"/>																																
<b>Save</b>		<b>New</b>	<b>Save Copy</b>	<b>Define Filters</b>	<b>Map Top-Level Filter</b>	<b>Dump SQL</b>	<b>Delete Setup</b>	<b>Help</b>																											
<a href="#">Report Setup</a>   <a href="#">Attachment &amp; Email</a>   <a href="#">Additional Attachment &amp; Email</a>   <a href="#">Output Options</a>   <a href="#">Digital Signature</a>																																			
<table border="1"> <thead> <tr> <th>Order</th> <th>Report Code</th> <th>Template File</th> <th>Script File</th> <th>Page Break Column</th> <th>Map Script Filter</th> <th>Sections</th> <th>Inactive?</th> <th>Delete?</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>glatco</td> <td>GLAttComp.xlsx</td> <td>CFheader.txt</td> <td></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </tbody> </table>									Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?	1	glatco	GLAttComp.xlsx	CFheader.txt		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Order	Report Code	Template File	Script File	Page Break Column	Map Script Filter	Sections	Inactive?	Delete?																											
1	glatco	GLAttComp.xlsx	CFheader.txt		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																											
					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																											

## 5 Define the top-level report filter fields. (**Define Filters** button)

**Report Filters Setup**

Merged Report: CFA GL Attribute Comparison report for YSR (cfa\_glatto)

**Save** **Preview** **Help**

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	List Values	Code to ID
Property	Property	1	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>		
CFType	CF Type	2	List		<input type="checkbox"/>	Attribute Comparison	
PostMonth	Post Month	3	Post Month		<input type="checkbox"/>		
Book	Book	4	Lookup List	ysiBookLookup	<input type="checkbox"/>		
SuppressZero	SuppressZero	5	Checkbox		<input type="checkbox"/>		
Template	Template	6	List		<input type="checkbox"/>	GLAttCo	
AttributeSortName	Summarize By	7	Lookup List		<input type="checkbox"/>	SELECT A.sname codeField	SELECT A.Hmy
					<input type="checkbox"/>		
					<input type="checkbox"/>		
					<input type="checkbox"/>		

### CFType

Optional. Used here as a display field. By including the CFA type here, you can retrieve the associated list value (Attribute Comparison) with another mini script. This can be useful, for example, if you want to display the CFA report type in your report header. If you add this field and refer to it in a script, you must map it (on the filter mapping screen associated with the script).

### Template

Included here not for retrieving the user's selection, but so that the column headers script can refer to the template scode (glatto, defined in the List Values column).

### AttributeSortName

Required. You must add a filter field that supplies Voyager with an attribute for summarizing data.

There are multiple ways to supply attribute names. In this example, a lookup list is defined as follows:

**List Values** SELECT A.sname codeField, A.sName TextField  
FROM AttributeName A WHERE 1=1

**Code to ID** SELECT A.Hmy IDField FROM AttributeName A  
WHERE 1=1 and A.sname IN ([CodeField])

- 6** Add report sections. Property Comparison reports require at minimum one section for retrieving data from Custom Financials and one scripted section for retrieving column headers.

Report Sections Setup

YSR Report:	CFA GL Attribute Comparison report for YSR (cfa_glatco)																																
Report Code:	glatco																																
<input type="button" value="Save"/> <input type="button" value="Close"/> <input type="button" value="Help"/>																																	
<input checked="" type="radio"/> Report Sections <input type="radio"/> Attachment Setup																																	
 <table border="1"> <thead> <tr> <th>Section Code</th> <th>Description</th> <th>SELECT Name</th> <th>Standard Analytics Report</th> <th>Map Standard Analytics Filter</th> <th>Tokens</th> <th>Relations</th> <th>Key Columns</th> </tr> </thead> <tbody> <tr> <td>CF</td> <td>Custom Financials</td> <td></td> <td>Custom Financials</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> <tr> <td>ColHead</td> <td>Column Headers</td> <td>Attrhead</td> <td></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </tbody> </table>		Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns	CF	Custom Financials		Custom Financials	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ColHead	Column Headers	Attrhead		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Section Code	Description	SELECT Name	Standard Analytics Report	Map Standard Analytics Filter	Tokens	Relations	Key Columns																										
CF	Custom Financials		Custom Financials	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																										
ColHead	Column Headers	Attrhead		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																										
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																										

- 7** Map filters for the Custom Financials data source.

Map Standard Analytics Filter

YSR Report:	CFA GL Attribute Comparison report for YSR (cfa_glatco)																																	
Report Code:	glatco																																	
SectionCode:	CF																																	
<input type="button" value="Save"/> <input type="button" value="Close"/> <input type="button" value="Help"/>																																		
 <table border="1"> <thead> <tr> <th>Standard Analytics Report Filters</th> <th>YSR Report Filters</th> <th>Constant Value</th> </tr> </thead> <tbody> <tr> <td>Template</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/> GLAttCo</td> </tr> <tr> <td>PropertyCode</td> <td><input type="checkbox"/> Property</td> <td><input type="checkbox"/></td> </tr> <tr> <td>FromMMYY</td> <td><input type="checkbox"/> PostMonth</td> <td><input type="checkbox"/></td> </tr> <tr> <td>ToMMYY</td> <td><input type="checkbox"/> PostMonth</td> <td><input type="checkbox"/></td> </tr> <tr> <td>BookCodeList</td> <td><input type="checkbox"/> Book</td> <td><input type="checkbox"/></td> </tr> <tr> <td>SuppressZero</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/> 1</td> </tr> <tr> <td>AttributeSortName</td> <td><input type="checkbox"/> AttributeSortName</td> <td><input type="checkbox"/></td> </tr> <tr> <td>AttributeSortValue</td> <td><input type="checkbox"/> AttributeSortName</td> <td><input type="checkbox"/></td> </tr> <tr> <td>TreeCode</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/> ysi_bs</td> </tr> <tr> <td>Detail</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/> 1</td> </tr> </tbody> </table>		Standard Analytics Report Filters	YSR Report Filters	Constant Value	Template	<input type="checkbox"/>	<input type="checkbox"/> GLAttCo	PropertyCode	<input type="checkbox"/> Property	<input type="checkbox"/>	FromMMYY	<input type="checkbox"/> PostMonth	<input type="checkbox"/>	ToMMYY	<input type="checkbox"/> PostMonth	<input type="checkbox"/>	BookCodeList	<input type="checkbox"/> Book	<input type="checkbox"/>	SuppressZero	<input type="checkbox"/>	<input type="checkbox"/> 1	AttributeSortName	<input type="checkbox"/> AttributeSortName	<input type="checkbox"/>	AttributeSortValue	<input type="checkbox"/> AttributeSortName	<input type="checkbox"/>	TreeCode	<input type="checkbox"/>	<input type="checkbox"/> ysi_bs	Detail	<input type="checkbox"/>	<input type="checkbox"/> 1
Standard Analytics Report Filters	YSR Report Filters	Constant Value																																
Template	<input type="checkbox"/>	<input type="checkbox"/> GLAttCo																																
PropertyCode	<input type="checkbox"/> Property	<input type="checkbox"/>																																
FromMMYY	<input type="checkbox"/> PostMonth	<input type="checkbox"/>																																
ToMMYY	<input type="checkbox"/> PostMonth	<input type="checkbox"/>																																
BookCodeList	<input type="checkbox"/> Book	<input type="checkbox"/>																																
SuppressZero	<input type="checkbox"/>	<input type="checkbox"/> 1																																
AttributeSortName	<input type="checkbox"/> AttributeSortName	<input type="checkbox"/>																																
AttributeSortValue	<input type="checkbox"/> AttributeSortName	<input type="checkbox"/>																																
TreeCode	<input type="checkbox"/>	<input type="checkbox"/> ysi_bs																																
Detail	<input type="checkbox"/>	<input type="checkbox"/> 1																																



Map the **Book** filter field to the **BookCodeList** element in the **Standard Analytics Report Filters** column. If you fail to specify a book for the **BookListCode** element, CFA defaults to the Cash book.

**8 Map filters for the mini script (return to the YSR Report Setup screen).**

**Map Script Filter**

YSR Report: CFA GL Attribute Comparison report for YSR (cfa\_glatco)  
Report Code: glattco

**Save    Close    Help**

YSR Report Filters		Script Filter Condition
Property (#Condition1#)		Property.Hmy in (#Property#)
Template (#Condition6#)		#Condition6#
AttributeSortName (#Condition7#)		AN.hmy in (#AttributeSortName#)

<b>Property (#Condition1#)</b>	Required. Establishes a mapping between the top-level Property filter field and the script that retrieves column headers.
<b>Template (#Condition6#)</b>	Required in this example report because the script that retrieves data for column headings uses a token to refer to the CFA template code. By using a token, the script remains flexible and can be adopted for other CFA-based reports.  Alternatively, you can hard-code a CFA template code into the script you use for retrieving column headings. The template code is necessary in the script in order to retrieve the column headings (e.g., MTD, YTD) appropriate to the CFA report template.
<b>AttributeSortName</b>	Required in this example report in order to pass the user's selection to the script that retrieves column headings.

**9 Generate your YSR report.**

**CFA GL Attribute Comparison report for YSR**

Property	comoff01^comoff02	Report Name	CFA GL Attribute Comparison report for YSR (cfa_glatco)																																																								
CF Type	Attribute Comparison	Output Type	Screen																																																								
Post Month	01/2017	Merge Reports	<input type="checkbox"/>																																																								
Book	Accrual	Show Grid	<input type="checkbox"/>																																																								
SuppressZero	<input checked="" type="checkbox"/>	Attach Reports	<input type="checkbox"/>																																																								
Template	GLAttCo	Email Reports	<input type="checkbox"/>																																																								
Summarize By	Market	Publish To SharePoint	<input type="checkbox"/>																																																								
		Show on Portal	<input type="checkbox"/>																																																								
		<b>Generate</b>	<b>Clear</b>																																																								
<input type="radio"/> Sheet1 <input checked="" type="radio"/> Sheet2																																																											
<b>Attribute Comparison Report</b> <table border="1"> <thead> <tr> <th>Acct Code</th> <th>Acct Desc</th> <th>Raleigh/Durham MTD</th> <th>Raleigh/Durham YTD</th> <th>Phoenix MTD</th> <th>Phoenix YTD</th> <th>Total MTD</th> <th>Total YTD</th> </tr> </thead> <tbody> <tr> <td>1000-0000</td> <td>ASSETS</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>1001-0000</td> <td>CURRENT ASSETS</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>1100-0000</td> <td>CASH &amp; CASH EQUIVALENT</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>1110-0000</td> <td>Cash - Operating</td> <td>\$ 228,670.00</td> <td>\$ 228,670.00</td> <td>\$ 158,650.00</td> <td>\$ 158,650.00</td> <td>\$ 387,320.00</td> <td>\$ 387,320.00</td> </tr> <tr> <td>1190-0000</td> <td>TOTAL CASH &amp; CASH EQUIVALENT</td> <td>\$ 228,670.00</td> <td>\$ 228,670.00</td> <td>\$ 158,650.00</td> <td>\$ 158,650.00</td> <td>\$ 387,320.00</td> <td>\$ 387,320.00</td> </tr> <tr> <td>1499-0000</td> <td>TOTAL CURRENT ASSETS</td> <td>\$ 228,670.00</td> <td>\$ 228,670.00</td> <td>\$ 158,650.00</td> <td>\$ 158,650.00</td> <td>\$ 387,320.00</td> <td>\$ 387,320.00</td> </tr> </tbody> </table>				Acct Code	Acct Desc	Raleigh/Durham MTD	Raleigh/Durham YTD	Phoenix MTD	Phoenix YTD	Total MTD	Total YTD	1000-0000	ASSETS							1001-0000	CURRENT ASSETS							1100-0000	CASH & CASH EQUIVALENT							1110-0000	Cash - Operating	\$ 228,670.00	\$ 228,670.00	\$ 158,650.00	\$ 158,650.00	\$ 387,320.00	\$ 387,320.00	1190-0000	TOTAL CASH & CASH EQUIVALENT	\$ 228,670.00	\$ 228,670.00	\$ 158,650.00	\$ 158,650.00	\$ 387,320.00	\$ 387,320.00	1499-0000	TOTAL CURRENT ASSETS	\$ 228,670.00	\$ 228,670.00	\$ 158,650.00	\$ 158,650.00	\$ 387,320.00	\$ 387,320.00
Acct Code	Acct Desc	Raleigh/Durham MTD	Raleigh/Durham YTD	Phoenix MTD	Phoenix YTD	Total MTD	Total YTD																																																				
1000-0000	ASSETS																																																										
1001-0000	CURRENT ASSETS																																																										
1100-0000	CASH & CASH EQUIVALENT																																																										
1110-0000	Cash - Operating	\$ 228,670.00	\$ 228,670.00	\$ 158,650.00	\$ 158,650.00	\$ 387,320.00	\$ 387,320.00																																																				
1190-0000	TOTAL CASH & CASH EQUIVALENT	\$ 228,670.00	\$ 228,670.00	\$ 158,650.00	\$ 158,650.00	\$ 387,320.00	\$ 387,320.00																																																				
1499-0000	TOTAL CURRENT ASSETS	\$ 228,670.00	\$ 228,670.00	\$ 158,650.00	\$ 158,650.00	\$ 387,320.00	\$ 387,320.00																																																				

## CHAPTER 7

# Working With Smart Markers and Excel

### In this chapter:

Smart Marker Overview .....	202
Smart Markers and Excel Data Types .....	203
Smart Marker Parameters .....	204
Excel Formulas and References .....	206
Groups and Subtotals .....	210
Images .....	220
Details (Show, Hide, and Delete with YSROptions).....	222
Charts, Graphs, and Pivot Tables .....	227
Tips and Tricks .....	233

This chapter describes how to write smart markers. Topics include using smart markers to format data and perform calculations and tips and tricks for working with Excel.



This section assumes you are working with Excel 2016, with additional information about Excel 2010 and 2013 for the topics on charts, graphs, and pivot tables.

## Smart Marker Overview

When you set up report templates for use with YSR, you can take advantage of all the formatting and data processing power of Excel. You can perform calculations, group and sort data, merge cells, skip rows, arrange data in alternating rows, insert images, and more. You can accomplish all these tasks with smart markers.

Smart markers are placeholders for Voyager data. All smart markers start with **&=**. If a smart marker results in more than one datum, then Voyager inserts extra rows into the report template to make room for the additional data (assuming the section associated with the smart marker has **Multiple Rows** selected on the **Report Sections Setup**). Thus you do not need to leave extra room in your report template to accommodate dynamically generated data; Voyager inserts extra rows for you.

## Smart Marker Syntax

YSR uses the following smart marker syntax:

&=SectionCode.FieldName

&=[SectionCode].[Field Name]



Brackets are optional unless a name contains an embedded space. Then they are required.

For example, &=Party.FullName is a smart marker that represents the data retrieved by the Party report section, in the FullName column.

## Smart Markers and Excel Data Types

Smart markers preserve the formatting you apply to them in Excel, and in many cases (dates, for example), you must apply data-type formatting in Excel. Use caution when applying data formats, however. The first row in any Excel report template must have the **General** data type. Apply date and other formatting to subsequent rows only.

Row 1 requires General data type format

Dates (row 2 and following) require date-type format

Failure to apply the **General** data type to row 1 can cause YSR reports to return no data.

# Smart Marker Parameters

## In this section:

Supported Parameters .....	204
Using Parameters to Display Data in Alternating Rows .....	205

This section shows how to use *parameters* to manipulate data arrangement in Excel. For example, you can use parameters to skip rows, arrange data in ascending or descending order, or expand data horizontally instead of vertically.

## Supported Parameters

You can add one or multiple parameters to the same smart marker. Separate them by commas, and do not add a space. For example, &=Party.FullName(copystyle,skip:1) copies the base cell's style to all cells in the column and skips one row for each row of data.



Parameters are not case-sensitive.

YSR supports the following parameters:

**noadd** Prevents Voyager from adding rows to accommodate expanding row data (limits results to the first result), unless used in combination with the **skip** parameter. Use **noadd** and **skip** together to display data on alternating rows. See "Using Parameters to Display Data in Alternating Rows" on page 205. Supported for the first worksheet in Excel workbooks only.

**skip:n** Skips **n** number of rows for each row of data (inserts **n** blank rows).

**ascending:n** Sorts data in ascending order, using **n** as the sort order key. For example, if you want to sort a set of names by LastName and then by FirstName, use the parameters (ascending:1) in the LastName column and (ascending:2) in the FirstName column.

**descending:n** Sorts data in descending order, using **n** as the sort order key.

**horizontal** Causes Voyager to insert additional data horizontally (using extra columns, left-to-right) rather than vertically (using extra rows). For an example and tips on working with horizontal data, see "Displaying Data Horizontally" on page 239.

**numeric** Converts text values to numbers where possible.

**shift** Shifts data down or right, creating extra rows or columns to fit data.



The shift parameter works the same way as related commands in Excel. For example, when you select a range of cells in Excel, right-click, select **Insert**, and select **Shift cells right** or **Shift cells down**, Excel shifts the data accordingly. The shift parameter functions identically for standard smart markers (which expand vertically) and horizontal smart markers (which expand in columns left-to-right).

**copystyle** Copies the base cell's style to all the cells in that column. Use this parameter, for example, to copy date formatting from one cell to the next.

**HTML** Indicates that the smart marker represents text data formatted with HTML tags. Voyager displays the formatted text. This is useful, for example, if you want to retrieve and display HTML-formatted text stored in the Memo or other notes table.

**group:normal** Establishes a sub-group of data. For example, you might group commercial lease data by property or owner so that you can calculate subtotals by property or owner. Add **:normal** to display the group identifier (the property code or owner code, for example) just once, rather than repeating the codes. For an illustration, see "Data Groups" on page 211.

**group:merge** Establishes a sub-group of data, useful for calculating subtotals by group. Add **:merge** to merge the cells containing the repeated element and show just one cell per group value (one cell per property or owner, for example). For an illustration, see "Data Groups" on page 211.

**group:repeat** Establishes a sub-group of data but does not eliminate multiple appearances of repeating group values. Useful where you want to add a sub-group for the purpose of calculating a subtotal without merging group values. For an illustration, see "Data Groups" on page 211.

## Using Parameters to Display Data in Alternating Rows

You can combine the **noadd** and **skip** parameters to display data in alternating rows. Add **noadd** on the first smart marker.

### Example Template

	A	B	C
1	&=Contact.Name(skip:1,noadd)		
2	&=Contact.Email(skip:1)		
3			

### Report Output

	A	B	C
1	John Doe		
2	<a href="mailto:john.doe@sample.com">john.doe@sample.com</a>		
3	Jane Doe		
4	<a href="mailto:jane.doe@sample.com">jane.doe@sample.com</a>		
5			

If you do not add the **noadd** parameter, Voyager displays all of the first data followed by all of the second data, rather than interspersing them.

## Excel Formulas and References

### In this section:

Example Formula - Running Balance.....	207
Formula Syntax and Parameters .....	207
Troubleshooting Broken Excel References .....	209
Troubleshooting Formulas That Display No Data .....	210

This section shows how to format Excel formulas and references to accommodate expanding row data.

In some cases, when working with Excel, you do not need to make any changes to your Excel formulas and references to capture dynamic YSR data. For example, the following report template has an Excel formula in cell D3:

	A	B	C	D
1	Tenant	Unit	Date	Monthly Rent
2	&=charges.Tenant	&=[charges].[UnitCoc	&=[charges].[Da	&=[charges].[MonthlyRent]
3				=SUM(D2)
4				

The rendered report sums the dynamically expanded data in column D, as expected:

Tenant	Unit	Date	Monthly Rent
Roxio, Inc. 350A	350A	12/31/2008	\$51,513.00
Roxio, Inc. 350B	350B	12/31/2008	\$51,513.00
Roxio, Inc. 350C	350C	12/31/2008	\$51,513.00
Roxio, Inc. 350D	350D	12/31/2008	\$51,513.00
			<b>\$206,052.00</b>

In most cases, however, you must re-format Excel formulas and references inside smart markers so that they refer correctly to dynamically expanded cells or ranges of cells. Smart markers that contain Excel formulas are called *dynamic formulas* and *repeating dynamic formulas*.



The behavior of Excel formulas varies according to the version of Excel used to design the report and the Correspondence Plug-in. Starting with Correspondence Plug-in 7.5, formulas that worked previously may need to be replaced with repeating dynamic formulas.

References to a cell (like =C5) may need to be replaced with dynamic references (&=&C5).

For more information, see "Troubleshooting Broken Excel References" on page 209.

- Dynamic formula smart markers begin with **&=** and render once.
- Repeating dynamic formula smart makers begin with **&=&=** and render as many times as indicated by the expanding row data.



Use a repeating dynamic formula when you want to perform the same calculation on multiple rows. For example, if you enter &=&=A{r}+B{r} in cell C1, Voyager sums the contents of cell A1 and B1 and displays the result in cell C1. Voyager also displays the sum of A2 and B2 in cell C2, the sum of A3 and B3 in cell C3, and so on.

## Example Formula - Running Balance

You can use formula smart markers to provide a running balance of dynamically expanding row data.

The following graphic shows a report template that uses both types of dynamic formulas (repeating and non-repeating) to display running totals and closing balance.

	A	B	C	D	E
1	DATE	CTRL NUM	DESCRIPTION	AMOUNT	RUNNING BALANCE
2					
3	&=stmt.TranDate	&=stmt.TranCtrl	&=stmt.TranDec	&=stmt.TranAmt	&=SUM(D{r},E{-1})
4			CLOSING BALANCE		&=E3
5					

Dynamic Formula

Repeating Dynamic Formula

Here is the rendered report:

	A	B	C	D	E
1	DATE	CTRL NUM	DESCRIPTION	AMOUNT	RUNNING BALANCE
2					
3	2/1/2014		Opening Balance	500	500
4	2/5/2014	C-156	Rent Charges	1200	1700
5	2/9/2014	C-181	Water Charges	165	1865
6	2/13/2014	R-95	Receipt (Check #02342)	-1500	365
7	2/17/2014	C-222	Common charges	40	405
8			CLOSING BALANCE		405
9					

The repeating dynamic formula in this example, `&=SUM(D{r},E{-1})`, uses supported syntax to refer to expanding row data. Note:

**D{r}** refers to whatever is in column D, in the same row (row r) as the rendered smart marker, after generating the report.

**E{-1}** refers to whatever is in column e, one row above the rendered smart marker, after generating the report.

For more information about formula syntax, see “Formula Syntax and Parameters” on page 207.

## Formula Syntax and Parameters

This section describes how to refer to data whose position is unknown at design-time and how to add parameters to formula smart markers.

## Row reference and offsets

Use curly braces to refer to a row number that is unknown at design time. For example, use `{r}` to refer to the row number on the same row as the rendered smart marker, after generating the report. Other examples include:

- `{-2}` Offset to two rows above the current row
- `{-1}` Offset to the row above the current row
- `{r}` Current row number
- `{1}` Offset to the row following the current row
- `{2}` Offset to two rows following the current row

## Column references

If you use the (horizontal) parameter to display data horizontally rather than vertically, you may find it useful to write smart markers that refer to unknown column letters. Use `{c}` to refer to a column letter not known at design time. For an example, see "Displaying Data Horizontally" on page 239.

## Adding parameters to formula smart markers

You can add parameters to formula smart makers by using the tilde (~) to separate the formula from the parameter. Place the parameter in parentheses.

Example: `&=&=A{r}/B{r}~(skip:1)`

This example is a repeating dynamic formula. It divides the value in the current row (r) of column A by the value in the current row (r) of column B and skips a row between each entry.

## Troubleshooting Broken Excel References

Starting with Correspondence Plug-in 7.5, formulas and references that worked previously may generate errors like #REF! or #VALUE!. These errors are due to changes in a third-party formula processing engine that supports MS Office 2016 file formats and file processing optimizations. These errors are encountered when all of the following apply:

- The report template is an Excel 2016 file.
- Voyager is using Correspondence Plug-in 7.5 or following.
- The problem smart marker is in a multi-row YSR report section.

The following grid provides examples of formulas and references that may break and how to fix them.

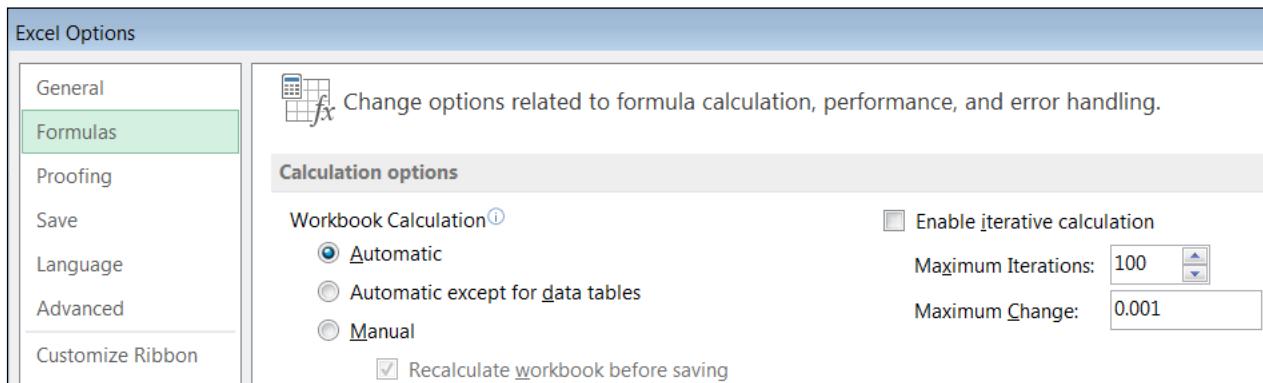
Type of Reference	Example	Possible Solutions
Reference to single cell	=G10	=SUM(G10:G10) &=&=G10
Reference to a cell inside a formula	=IF(N9=1,"Yes","No")	&=&=(N{-2}=1,"Yes","No") <b>TIP</b> Replace the reference to row 9 with a reference to the position of the cell, offset from the smart marker (indicated by {-2} in this example).
All components of a formula	=IF(N9=1,M9,-SUM(H9:H9)+SUM(G9:G9))	&=&=IF(\$N{-2}=1,\$M{-2},-SUM(\$H{-2}:\$H{-2})+SUM(\$G{-2}:\$G{-2})) <b>TIP</b> Add the &=&= prefix to the whole smart marker; add the \$ prefix to each column letter, and replace references to row numbers with dynamic offsets {-2} in this example).

## Troubleshooting Formulas That Display No Data

If automatic formula calculations are turned off in Excel, calculations do not execute when you generate a report on screen or publish to Excel or PDF. If you publish to Excel and click into one of the cells containing a formula, you can see (in the formula bar on the Excel ribbon) that the formula is still there, and it has been modified by Voyager to encompass expanded row references. However, no data appears.

In this situation, make sure that automatic calculations are turned off.

In Excel, select **File > Options > Formulas > Calculation Options > Workbook Calculation**. Make sure that **Automatic** is selected.



## Groups and Subtotals

### In this section:

Data Groups .....	211
Subtotaling a Single Data Group .....	212
Subtotaling Multiple Data Groups .....	213
Labeling Subtotals .....	215
Grand Total Tricks .....	218

This section describes how to group your data and calculate subtotals by group.

You can provide multiple subtotals in the same report template. If you are working with commercial lease data, for example, you might want to provide subtotals of financial data by tenant (lease), by property, and by owner. You can automatically subtotal any section of data that you can identify by group.

## Data Groups

To establish data groups, add a grouping parameter to one or more smart markers. For example, if you want to group financial data by property, use add the grouping parameter to your property smart marker.

Voyager supports the following grouping parameters:

### **group:normal**

&=SectionCode.FieldName(group:normal)

	A	B	C	
1	Order ID	Quantity	UnitPrice	
2	&=[Order Details].OrderID(group:normal)	&=[Order Details].Quantity	&=[Order Details].UnitPrice	
3			&=subtotal9:Order Details.OrderID	
4				
5	A	B	C	
1	Order ID	Quantity	UnitPrice	
2		10248	12	\$14.00
3			10	\$9.80
4			5	\$34.80
5				\$58.60
6		10249	9	\$18.60
7			40	\$42.40
8				\$61.00

### **group:merge**

&=SectionCode.FieldName(group:merge)

	A	B	C	
1	Order ID	Quantity	UnitPrice	
2	&=[Order Details].OrderID(group:merge)	&=[Order Details].Quantity	&=[Order Details].UnitPrice	
3			&=subtotal9:Order Details.OrderID	
4				
5	A	B	C	
1	Order ID	Quantity	UnitPrice	
2			12	\$14.00
3			10	\$9.80
4		10248	5	\$34.80
5				\$58.60
6			9	\$18.60
7		10249	40	\$42.40
8				\$61.00

**group:repeat**

`&=SectionCode.FieldName(group:repeat)`

Useful when you want to establish a data group but you do not want to eliminate duplicate group values in your report.

	A	B	C	
1	Order ID	Quantity	UnitPrice	
2	&=[Order Details].OrderID(group:repeat)	&=[Order Details].Quantity	&=[Order Details].UnitPrice	
3			&=subtotal9:Order Details.OrderID	
4				
5	A	B	C	
1	Order ID	Quantity	UnitPrice	
2		10248	12	\$14.00
3		10248	10	\$9.80
4		10248	5	\$34.80
5				\$58.60
6		10249	9	\$18.60
7		10249	40	\$42.40
8				\$61.00



You can combine grouping parameters with other parameters, as normal. For example, `&=[Order Details].OrderID(group:normal,skip:1)` groups order data by order ID and skips one row after each group.

## Subtotaling a Single Data Group

You can calculate subtotals for a group of data by using the **subtotalN** parameter. N represents numbers between 1 and 11, corresponding to the various subtotal functions available in Excel, as follows:

1	AVERAGE
2	COUNT
3	COUNTA
4	MAX
5	MIN
6	PRODUCT
7	STDEV
8	STDEVP
9	SUM
10	VAR
11	VARP



Refer to Microsoft Excel Help for more information about each of these functions.

**Subtotal smart marker format:**

&=subtotalN:SectionCode.FieldName

**Example:**

&=subtotal9:OrderDetails.OrderID

This smart marker represents the subtotal of data categorized by Order ID. For example, the following graphic shows a report template that uses a subtotal in cell C3 to sum the data in column C, by OrderID group (column A).

The screenshot shows a two-part Excel interface. The top part is a report template with columns A, B, and C. Row 1 has headers 'Order ID', 'Quantity', and 'UnitPrice'. Row 2 contains smart markers: '&=[Order Details].OrderID(group:merge)' in A2, '&=[Order Details].Quantity' in B2, and '&=[Order Details].UnitPrice' in C2. Row 3 is blank. Row 4 is also blank. Row 5 starts a data section with a header row: 'Order ID', 'Quantity', 'UnitPrice'. The data rows show two groups of items. Group 1 (rows 6-8) has OrderID 10248, Quantity 12, and UnitPrice \$14.00. Group 2 (rows 9-11) has OrderID 10249, Quantity 40, and UnitPrice \$18.60. Subtotals are shown in the last row of each group: '\$58.60' for group 1 and '\$61.00' for group 2.

A	B	C
1 Order ID	Quantity	UnitPrice
2 &=[Order Details].OrderID(group:merge)	&=[Order Details].Quantity	&=[Order Details].UnitPrice
3		&=subtotal9:Order Details.OrderID
4		
5		
A	B	C
1 Order ID	Quantity	UnitPrice
2		12
3		10
4	10248	5
5		\$34.80
6		\$14.00
7	10249	9
8		\$9.80
		\$58.60
		\$18.60
		\$42.40
		\$61.00

**Formatting and placement of subtotalN smart markers**

- Place the subtotalN smart marker in the row immediately following the data you want to summarize.
- The group and grouped data must appear on the same row in the Excel template before rendering. (For example, &=OrderDetails.OrderID and &=OrderDetails.UnitPrice appear in the same row.)
- The field name used in the subtotalN smart marker is case sensitive within Excel. Use the same casing in the subtotalN smart marker as you use in the smart marker for the data group. (For example, OrderID appears in the smart markers in cell A2 and cell C3.)

## Subtotaling Multiple Data Groups

In some cases, you may want to provide subtotals for multiple groups of data within the same report. For example, you might create an Aging Detail Report that displays charges by individual lease, by property, and by property owner. In the same report, you can calculate and display the subtotal of charges owed on each lease, the subtotal for each property, and the subtotal for each property owner.

To calculate and display multiple subtotals, create a single subtotal smart marker and use "&" to list additional groups for subtotaling.

**Example:** &=subtotal9:Aging.ownercode&Aging.propcode&Aging.leasecode



List the groups you want to subtotal by section code and field name (for example, Aging.ownercode). Subtotal smart markers must refer to grouped items that appear elsewhere in the same report.

To add subtotals to your report, enter the subtotal smart marker in the cell immediately following the values you want to subtotal.

F8	f <sub>x</sub>	&=subtotal9:Aging.ownercode&Aging.propcode&Aging.leasecode				
	A	B	C	D	E	F
<b>Aging Detail Report</b>						
2						
3	Owner: &=HeaderFooter.Owner Name (&=Aging.ownercode(group:normal))					
4						
5	Owner Code	Property	Charge To	Invoice No	Charge Code	Total Charges
7	&=Aging.ownercode(group:parent)	&=Aging.propcode(group:parent)	&=Aging.leasecode(group:parent)	&=Aging.InvoiceNo	&=Aging.ChargeCode	&=Aging.Charge_Amount
8						&=subtotal9:Aging.ownercode&Aging.propcode&Aging.leasecode
9						

This template generates a report with subtotals by owner, property, and individual lease.

	A	B	C	D	E	F
2	<b>Aging Detail Report</b>					
3	Owner: Abbie Greisman (nlow0001)					
4						
5	Owner Code	Property	Charge To	Invoice No	Charge Code	Total Charges
7	nlow0001	nlp00001	t0001701	201200000006 201200000006 201200000009 201200000009	rent misc rent misc	5,500.00 3,150.00 5,500.00 3,150.00
8						
9						
10						
11						<b>17,300.00</b>
12						<b>17,300.00</b>
13						
14		nlp00002	t0001702	201200000007 201200000007	rent appfee	11,000.00 1,100.00
15						
16						<b>12,100.00</b>
17			t0001709	201200000028 201200000028 201200000028	rent capital fasb	5,500.00 7,700.00 1,100.00
18						
19						
20						<b>14,300.00</b>
21			t0001702	201200000010 201200000010	rent appfee	11,000.00 1,100.00
22						
23						<b>12,100.00</b>
24			t0001709	201200000032 201200000032 201200000032	rent capital fasb	5,500.00 7,700.00 1,100.00
25						
26						
27						<b>14,300.00</b>
28						<b>52,800.00</b>
29						
30						<b>70,100.00</b>
31						



Voyager processes YSR templates from bottom to top. Therefore, Voyager displays the subtotal for the group that you specify first on the lowest line of the report. The next subtotal appears on the next line up, and so on. In this example, the owner's subtotal appears at the bottom of the report, preceded by the property subtotals, preceded by lease subtotals.



To make your report more readable, you can add labels for each subtotal. For more information, see “Labeling Subtotals” on page 215.

Concatenated subtotalN smart markers have the same formatting and placement requirements as single subtotalN smart markers. For more information, see “Formatting and placement of subtotalN smart markers” on page 213.

## Labeling Subtotals

You can add labels to one or more subtotals by using the **Label** and **LabelPosition** parameters. If your subtotal smart marker contains more than one data group, you must define the parameters separately for each group. For example, the following smart marker contains two data groups for subtotaling and provides label parameters for each:

`&=subtotal9:Aging.ownercode(Label:"Owner Total",LabelPosition:-2)&Aging.propcode(Label:"Property Total",LabelPosition:-1)`

### Label Parameter Formats

**Label:“Total for {0}”** Specifies the text that Voyager displays. **{0}** is an optional placeholder for the name of the group you are subtotaling.

- “Total” returns “Total”
- “Total for {0}” returns “Total for GroupName”
- “{0} Subtotal” returns “GroupName Subtotal”



**{0}** is a placeholder that refers only to the name of the group that you are subtotaling. For example, the smart marker “&=subtotal9:Report.Year” provides subtotals grouped by year. By appending the parameter (Label:“Total for {0}”) you can generate labels like “Total for 2010,”“Total for 2011,”“Total for 2012,” and so on.

**LabelPosition:N** Specifies the column offset for the label.

- “LabelPosition:-2” positions the label two columns to the left of smart marker.
- “LabelPosition:1” positions the label one column to the right of the smart marker.



The LabelPosition parameter is optional. By default, Voyager places the label one column to the left (-1) of the smart marker.

### Example report with subtotal for a single data group

The following graphic shows an Excel template that provides subtotals of report data by year.

Cell D3 contains this smart marker:

`&=subtotal9:Report.Year(Label:"Total",LabelPosition:-3)`

	A	B	C	D	E
1	Year	Date	Asset Class	Reported Cost	Assessed Value
2	<code>&amp;=Report.Year(Group:Repeat,skip:1)</code>	<code>&amp;=Report.Date(copystyle)</code>	<code>&amp;=Report.[Asset Class](Group:Repeat)</code>	<code>&amp;=Report.[Reported Cost]</code>	<code>&amp;=Report.[Assessed Value]</code>
3				<code>&amp;=subtotal9:Report.Year(Label:"Total",LabelPosition:-3)</code>	
4					

Voyager calculates subtotals by year, adds the label "Total," and positions the label three columns to the left of the smart marker:

	A	B	C	D	E
1	Year	Date	Asset Class	Reported Cost	Assessed Value
2	2010	4/15/2010	Equipment	400	350
3	2010	8/15/2010	Inventory	800	600
4	2010	10/15/2010	Equipment	600	700
5	Total			1800	1650
6					
7	2009	4/15/2009	Inventory	300	350
8	2009	8/15/2009	Inventory	500	400
9	Total			800	750
10					

### Example report with subtotal for multiple data groups

The following graphic shows an Excel template that provides subtotals of outstanding charges by owner, property, and individual lease.

Cell F8 contains this smart marker:

```
&=subtotal9:Aging.ownercode(Label:"Total for {0}")&Aging.propcode(Label:"Total for {0}",LabelPosition:-4)&Aging.leasecode(Label:"Total for {0}",LabelPosition:-3)
```

	A	B	C	D	E	F
<b>Aging Detail Report</b>						
2						
3 Owner: &=HeaderFooter.Owner Name (&=Aging.ownercode(group:normal))						
5	Owner Code	Property	Charge To	Invoice No	Charge Code	Total Charges
7	&=Aging.ownercode	&=Aging.propcode(g	&=Aging.leasecode(	&=Aging.InvoiceNo	&=Aging.ChargeCode	ing.Charge_Amount
8	code(Label:"Total for {0}")&Aging.propcode(Label:"Total for {0}",LabelPosition:-4)&Aging.leasecode(Label:"Total for {0}",LabelPosition:-3)					
9						

This template generates results as follows, with subtotal labels in various columns:

	A	B	C	D	E	F
<b>Aging Detail Report</b>						
2						
3 Owner: Abbie Greisman (nlow0001)						
4						
5	Owner Code	Property	Charge To	Invoice No	Charge Code	Total Charges
7	nlow0001	nlp00001	t0001701	201200000006 201200000006 201200000009 201200000009	rent misc rent misc	5,500.00 3,150.00 5,500.00 3,150.00
8						
9						
10						
11			Total for t0001701			17,300.00
12		Total for nlp00001				17,300.00
13						
14		nlp00002	t0001702	201200000007 201200000007	rent appfee	11,000.00 1,100.00
15						
16			Total for t0001702			12,100.00
17						
18			t0001709	201200000028 201200000028 201200000028	rent capital fasb	5,500.00 7,700.00 1,100.00
19						
20			Total for t0001709			14,300.00
21						
22			t0001702	201200000010 201200000010	rent appfee	11,000.00 1,100.00
23			Total for t0001702			12,100.00
24						
25			t0001709	201200000032 201200000032 201200000032	rent capital fasb	5,500.00 7,700.00 1,100.00
26						
27			Total for t0001709			14,300.00
28		Total for nlp00002				52,800.00
29						
30					Total for nlow0001	70,100.00
31						

Labels for each subtotal appear in different columns, as defined by each LabelPosition parameter in the subtotal smart marker.

## Grand Total Tricks

In some cases you may want to calculate totals at different levels than the groups that are contained in your rendered report. For example, suppose you are creating an AR Aging report like the one illustrated in previous sections, but you do not want to (or cannot) include an owner column. Nevertheless, you want to provide a grand total of all the aging amounts for all the properties selected by the user.

In this case, you cannot simply insert an Excel formula like **=sum(F7)** to sum the total charges in column F, because column F contains both subtotals and raw data. In other words, you cannot do this:

Aging Detail Report					
3	Owner: &=HeaderFooter.Owner Name (&=Aging.ownercode(group:normal))				
4					
5	Property	Charge To	Invoice No	Charge Code	Total Charges
7	&=Aging.propcode(g)	&=Aging.leasecode(	&=Aging.InvoiceNo	&=Aging.ChargeCode	Aging.Charge_Amount
8	code(Label:"Total for {0}")&Aging.propcode(Label:"Total for {0}",LabelPosition:-4)&Aging.leasecode(Label:"Total for {0}",LabelPosition:-3)				
9					=sum(F7)

Instead, create a dummy data group at the level of the grand total, analogous to the owner level in this example. For example, if you are working with a script, retrieve an extra line of dummy data:

```
//SELECT Aging
Select
1 AS DummyGroup
,p.scode propcode
...
```

The result set for the script above has an extra column aliased **DummyGroup** that contains '1' in every row. You will use the dummy column of 1s as the highest-level data group for your report.

Add the results to your report template and group them. For example add a smart marker like the following:

&=Aging.DummyGroup(group:normal)

Hide the column that contains the fake data.

Last, use the **DummyGroup** as part of the **subtotalN** smart marker.

Modify these procedures as necessary to calculate subtotals of different data groups.

## Images

You can add images to your Excel templates by using a SQL statement to retrieve your images and using the **picture** parameter to place them in your template.

### Retrieving Images

To retrieve images, you must know the naming convention and location of your images. Ask your Voyager administration how your organization stores and names your images.

Suppose, for example, your organization stores images in the Pictures folder in the Reports path, using property scodes for filenames. Write a select statement that retrieves scodes and concatenates them with the image file location, as demonstrated in the following text:

```
//SELECT charges
Select
p.scode property
,p.saddr1 name
,Concat ('\\\' + ServerName + '\Reports\Pictures\' , ltrim(rtrim(p.scode)) , '.jpg')
AS PictureName
FROM property p
WHERE 1=1 #condition1#
//END SELECT
```



The SQL Server Concat function was introduced in SQL Server 2012. If your Voyager system runs against an earlier SQL Server database (2008 R2 or earlier), use a more basic method for assembling strings in T-SQL such as:

```
'\\\' + ServerName + '\Reports\Pictures\' + ltrim(rtrim(Cast(p.scode as nvarchar(8)))) + '.jpg' as PictureName
```

As a best practice, you can use the ltrim and rtrim functions to remove any blank spaces from the string.

If you store your images somewhere else than the Reports path, you must edit your select statement accordingly. For example, you might retrieve images from a webshare or an external site. Possible fragments of a select statement for images include:

```
Concat('https://www.yardixyz.com/ShareName/images/' , ltrim(rtrim(p.scode)) , '.jpg') AS PictureName
```

```
Concat('https://www.ExternalSite.com/images/' , ltrim(rtrim(p.scode)) , '.jpg') AS PictureName
```

Your SaaS level and security settings may affect your access to images.

### Placing Images in Report Templates

To add an image to an Excel template, add a smart marker that refers to the alias you used in your SQL select section to alias your images' file location. Then add the **picture** parameter to tell Voyager to replace the file location with the image itself.

For example, the following graphic shows an Excel template with an image in cell C2. The image is in the select section coded **charges** and is aliased **picturename**.

A	B	C
1	Property Scode	Property Name
2	&=[charges].[property]	&=[charges].[name]
3		&=charges.picturename(picture:scale100)
4		

Image Smart Marker



Starting with Correspondence Plug-in 7.5, you can use square brackets [ ] in image smart markers.

## Formatting Images

There are several ways to format images with the **picture** parameter. Formatting options include:

**picture:FitToCell** Auto-fits the image to the cell's height and width.

&=SectionCode.Alias(picture:FitToCell)

**picture:ScaleN** Scales the image to N percent of the original, preserving the image's aspect ratio.

&=SectionCode.Alias(picture:Scale80)

**picture:ScaleN&N** Scales the image to N percent of the height and N percent of the width of the original.

&=SectionCode.Alias(picture:Scale50&60)

**picture:Width:Nin&Height:Nin** Renders the image N inches wide and N inches high.

&=SectionCode.Alias(Picture:Width:2in&Height:3in)



**picture:Left:Npt&Top:Npt&Width:Nin&Height:Nin** Renders the image N inches wide and N inches high, and positions the image N points from the top and N points from the left of the cell.

&=SectionCode.Alias(Picture:Left:2pt&Top:2pt&Width:2in&Height:3in)



## Details (Show, Hide, and Delete with YSROptions)

You can give users the option to show or hide detail data in Excel-based YSR reports by taking advantage of YSROptions. YSROptions is a set of report-design conventions that enables users to show and hide column data as desired.

For example, the following graphic shows a YSR report filter with a **Show Details** check box. When the user selects **Show Details**, detail data appears in the report.

**Trans Register**

Trans Type	Journal	Report Name	Trans Register (TransReg)											
Legal Entity	alble	Output Type	Screen											
Property		Merge Reports	<input type="checkbox"/>											
Account		Show Grid	<input type="checkbox"/>											
From Date		Generate												
To Date		Clear												
Show Details	<input checked="" type="checkbox"/>	Show on Portal												
Show Property Name	<input type="checkbox"/>													
<input type="checkbox"/> Register Reports <input checked="" type="checkbox"/> Show Detail <input type="checkbox"/> YSROptions														
Transaction Register Trans Type: Journal Legal Entity: alble														
Doc. Seq. No.	CtrlNoID	CtrlNo	Period	Due Date	Invoice Date	From Date	To Date	Charge To	Batch	Control	Legal Entity	Property	Charge Code	B
10	1000000049	300000049	10/2007	10/17/2007				Both	-1399999956	C-300000049	ctown	ctprop		
10	1000000049	300000049	10/2007	10/17/2007				Both	-1399999956	C-300000049	ctown	ctprop		
13	1000000051	300000051	09/2007	10/17/2007				US Dollar	-299999944	C-300000051	ctown	ctprop		
13	1000000051	300000051	09/2007	10/17/2007				US Dollar	-299999944	C-300000051	ctown	ctprop		
200800000001	1000000064	300000064	03/2008	04/08/2008				Both	-299999931	C-300000064	dele	deprop		
200800000001	1000000064	300000064	03/2008	04/08/2008				Both	-299999931	C-300000064	dele	deprop		
200800000001	1000000065	300000065	03/2008	04/29/2008				Both	-1600000000	C-300000065	ukle	ukprop1		
200800000001	1000000065	300000065	03/2008	04/29/2008				Both	-1600000000	C-300000065	ukle	ukprop1		
200800000002	1000000071	300000071	06/2008	06/10/2008				Both	-1600000000	C-300000071	ukle	ukprop2		
200800000002	1000000071	300000071	06/2008	06/10/2008				Both	-1600000000	C-300000071	ukle	ukprop2		
200800000003	1000000072	300000072	06/2008	06/10/2008				Both	-299999927	C-300000072	ukle	ukprop1		
200800000003	1000000072	300000072	06/2008	06/10/2008				Both	-299999927	C-300000072	ukle	ukprop1		

Excel columns B-H

When the user clears the **Show Details** check box, Voyager hides detail data in the report.

The screenshot shows the 'Trans Register' report configuration screen. On the left, there are filters for 'Trans Type' (Journal), 'Legal Entity' (alble), 'Property', 'Account', 'From Date', 'To Date', 'Show Details' (unchecked), and 'Show Property Name' (unchecked). On the right, settings include 'Report Name' (Trans Register (TransReg)), 'Output Type' (Screen), 'Merge Reports' (unchecked), 'Show Grid' (unchecked), 'Attach Reports' (unchecked), 'Email Reports' (unchecked), and 'Show on Portal' (unchecked). Below these are 'Generate' and 'Clear' buttons. At the bottom, tabs for 'Register Reports', 'Show Detail', and 'YSROptions' are visible, with 'Show Detail' being the active tab. The main area displays a transaction register table with columns: Doc. Seq. No., Charge To, Batch, Control, Legal Entity, Property, Charge Code, Base Curr., Net Amount, Vat Amount, Gross Amount, Amount Paid, and Reference. The table contains several rows of transaction data. A callout box points to the first row with the text 'Excel columns B-H are hidden'.

Doc. Seq. No.	Charge To	Batch	Control	Legal Entity	Property	Charge Code	Base Curr.	Net Amount	Vat Amount	Gross Amount	Amount Paid	Reference
10	Both	-139999956	C-300000049	ctown	ctaprop		eur	0.00	0.00	0.00		:Currency G
10	Both	-139999956	C-300000049	ctown	ctaprop		eur	35.92	0.00	35.92		:Currency G
13	US Dollar	-299999944	C-300000051	ctown	ctaprop		eur	85.46	0.00	85.46		
13	US Dollar	-299999944	C-300000051	ctown	ctaprop		eur	0.00	0.00	0.00		
200800000001	Both	-299999931	C-300000064	dele	deprop		eur	0.00	0.00	0.00		
200800000001	Both	-299999931	C-300000064	dele	deprop		eur	100.00	0.00	100.00		
200800000001	Both	-1600000000	C-300000065	ukde	ukprop1		gbp	175.00	0.00	175.00		:VatXfer
200800000001	Both	-1600000000	C-300000065	ukde	ukprop1		gbp	0.00	0.00	0.00		:VatXfer
200800000002	Both	-1600000000	C-300000071	ukde	ukprop2		gbp	0.00	0.00	0.00		:VatXfer
200800000002	Both	-1600000000	C-300000071	ukde	ukprop2		gbp	17.50	0.00	17.50		:VatXfer
200800000003	Both	-299999927	C-300000072	ukde	ukprop1		gbp	1,000.00	0.00	1,000.00		

You can set up a report with an option to show or hide detail by including the following components in your report design:

- A **Show Details** check box in the top-level YSR report filter
- A smart marker in your Excel template that reflects the value of the **Show Details** check box (1 if true, 0 if false)
- A statement on the YSROptions tab of your Excel template that shows/hides details depending on the value of the smart marker. The YSROptions tab must respect the conventions described in the procedure that follows this topic.

## Named versus unnamed worksheets

The YSROptions report-design conventions require a reference from the YSROptions worksheet to column data in other worksheets of the report. For example, the following graphic shows, in column A, references to column data in the worksheet named **Register Reports**.

	A	B	C	D	E
1	ColumnAddress	Hide	Delete		
2	Register Reports!A				
3	Register Reports!B	Y			
4	Register Reports!C	Y			
5	Register Reports!D	Y			
6	Register Reports!E	Y			
7	Register Reports!F	Y			
8	Register Reports!G	Y			
9	Register Reports!H	Y			

◀ ▶ | Register Reports | Show Detail | **YSROptions**

Your Excel template might contain unnamed worksheets, however (i.e., worksheets with Excel's default names, Sheet1, Sheet2, and Sheet3). In such cases, Voyager replaces the worksheet name with the YSR report code and key column value at run time (for example, transreg-comoff01).

Therefore, you cannot simply refer to unnamed worksheets (Sheet1, Sheet2, or Sheet3) on the YSROptions worksheet. You must replace references to Sheet1 etc. with references to the rendered report worksheet names, followed by an underscore and the number of the target worksheet. For example:

	A	B	C	D
1	ColumnAddress	Hide	Delete	
2	transreg-comoff01_1!B	Y		
3	transreg-comoff01_1!C	Y		
4	transreg-comoff01_1!D	Y		
5	transreg-comoff01_2!B	Y		
6				
7				
8				

◀ ▶ | Sheet1 | Sheet2 | **YSROptions** | **+**

In this example, columns B, C, and D of sheet1 and column B of sheet2 are hidden.

### YSROptions limitations

You cannot hide or delete data in column A of any worksheet.

## To give users the option to show, hide, or delete column data

- 1 Add a **Show Details** filter field to the top-level YSR report filter. Make sure that **Type** is set to **Check-box**.

**Report Filters Setup**

Merged Report: Trans Register (TransReg)

Save Close

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?
TransType	Trans Type	01	List		<input type="checkbox"/>		<input type="checkbox"/>
OwnerId	Legal Entity	02	Lookup List	ysiLegalEntityLookup	<input checked="" type="checkbox"/>		<input type="checkbox"/>
PropertyId	Property	03	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>	02	<input type="checkbox"/>
Acct	Account	04	Lookup List	ysiAccountLookup	<input type="checkbox"/>		<input type="checkbox"/>
FromDate	From Date	08	Date		<input type="checkbox"/>		<input type="checkbox"/>
Todate	To Date	09	Date		<input type="checkbox"/>		<input type="checkbox"/>
ShowDetail	Show Details	10	Checkbox		<input type="checkbox"/>		<input type="checkbox"/>
ShowPropName	Show Property Name	11	Checkbox		<input type="checkbox"/>		<input type="checkbox"/>

- 2 Add a worksheet named **YSROptions** to your Excel report template.
- 3 Somewhere in your Excel report template, add the following smart marker:

&=YSRFilterValues.ShowDetail



YSRFilterValues is a supported section code. ShowDetail is the field name of the report filter field. If your filter field has a different field name, edit the smart marker accordingly.

For simplicity, add the smart marker to the YSROptions tab.

The &=YSRFilterValues.ShowDetail smart marker returns 1 if true and 0 if false.

- 4 Make sure that the worksheet that contains the data you want to show or hide has a name.

A	B	C	D	E	F	G	H	I
1	Transaction Register							
2	&=YSRFilterValues.ReportTitleValues							
4	Doc. Seq. No.	CtrlNoID	CtrlNo	Period	Due Date	Invoice Date	From Date	To Date
5	&=TransR.DocNo	&=TransR.I &=&=B(r)-70(&=Transf &=TransR.T &=TransR &=TransR.F &=TransR.T &=TransR.BookNam						
6								
10	Register Reports	Show Detail	YSROptions					



If you do not name your worksheet, YSR replaces the default worksheet name (Sheet1, for example) with the report section code and key column value of the rendered report at run time. This replacement breaks the reference between the YSROptions worksheet and the body of the report that you establish in the next step.

For a workaround to this problem, see “Named versus unnamed worksheets” on page 224.

- 5 On the YSROptions worksheet, add the following headers, in this order: **ColumnAddress**, **Hide**, **Delete**.

	A	B	C	D
1	ColumnAddress	Hide	Delete	

- 6 Complete the YSROptions worksheet. For example:

	A	B	C	D	E	F	G	H
1	ColumnAddress	Hide	Delete					
2	Register Reports!A							
3	Register Reports!B	Y						
4	Register Reports!C	Y						
5	Register Reports!D	Y						
6	Register Reports!E	Y						
7	Register Reports!F	Y						
8	Register Reports!G	Y						
9	Register Reports!H	Y						
10								

**Column A** Use this convention to refer to the column you want to show/hide:

SheetName!Column

**TIP** You cannot hide or delete data in column A of a worksheet.

**Column B (Hide)** Leave blank, enter **Y** or **N**, or use a formula to enter **Y** or **N** based on the user's selection at run time.

This example uses a formula that refers to cell F9, which contains a smart marker that represents the user's choice to show (1) or hide (0) details.

If F9 contains 1, then the Hide cell is set to N; If F9 contains 0, then the Hide cell is set to Y.

**Column C (Delete)** Leave blank, enter **Y** or **N**, or use a formula to enter **Y** or **N** based on the user's selection at run time.

**Cell F9** In this example, the smart marker representing the user's selection at run time is located in cell F9. You can place this smart marker anywhere in your Excel template.

- 7 Save your Excel report template to the Reports path.

- 8 Generate your report.

# Charts, Graphs, and Pivot Tables

This section describes how to create charts, graphs, and pivot tables with Excel report templates.



This section describes best practices when working with Excel 2010, 2013, or 2016. You can design a template in any of these versions, but you must save it as an .xls file (Excel 97-2003) in order for it to render in YSR.

## Adding Charts, Graphs, and Pivot Tables

Creating graphical reports for Excel report templates requires no special design procedures. To create a graphical element, use normal Excel design steps and make sure the element includes, as its data range (or ranges), a series of cells that will be populated at run time.

The easiest way to accomplish this is to start with dummy data, design the graphical element, replace the first line of your dummy data with a smart marker, and then delete the dummy data. Leave the data range expanded to include all the rows previously occupied by the dummy data.

The following topics describe best practices for creating a chart in Excel 2010, 2013, or 2016. Modify them as necessary to create graphs or pivot charts.

### This section includes the following procedures:

- |  |     |
|--|-----|
| To create a graph with Excel 2010.....         | 227 |
| To create a graph with Excel 2013 or 2016..... | 230 |

### To create a graph with Excel 2010

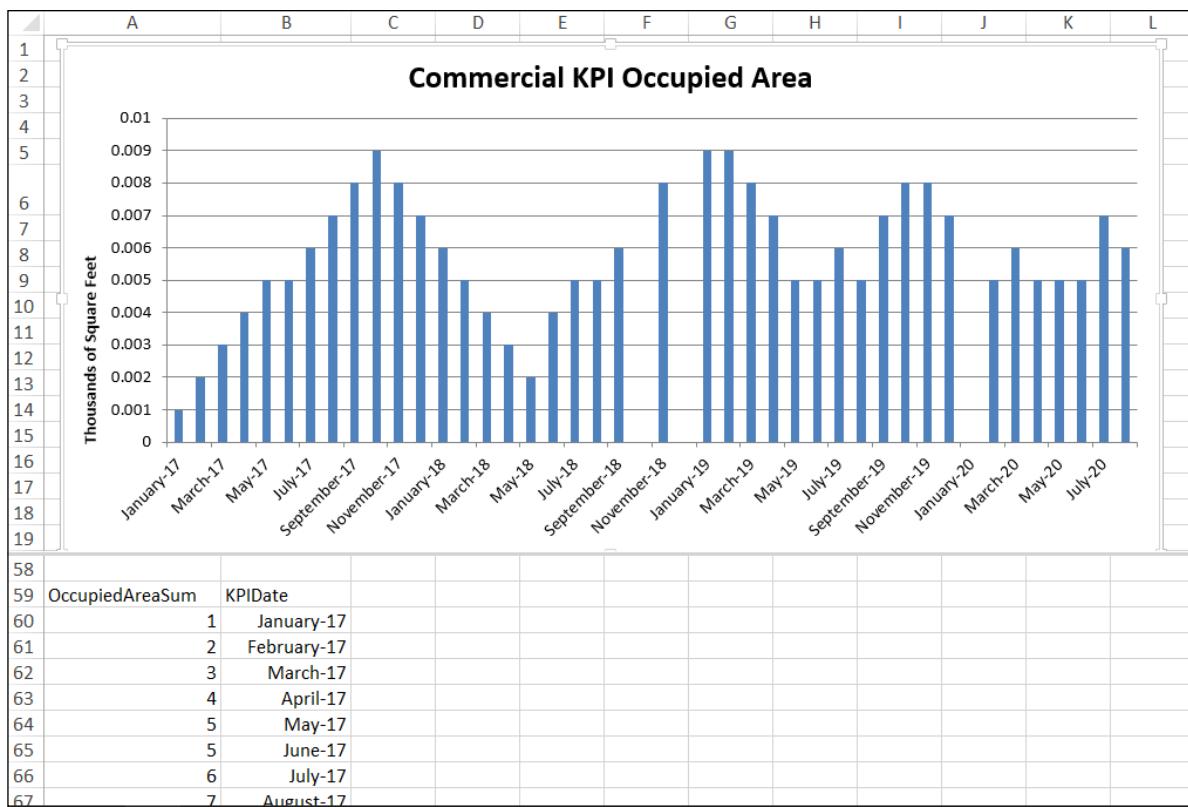
- 1 Add dummy data to your report template.

58						
59	OccupiedAreaSum	KPIDate				
60	1	January-17				
61	2	February-17				
62	3	March-17				
63	4	April-17				
64	5	May-17				
65	6	June-17				
66	7	July-17				
67		August-17				



Use dummy data that corresponds generally with the data you want to represent (dates for date ranges, numbers for numeric data, and so on).

- 2** Add a graph to your template, using the dummy data.



Use the dummy data as the source data for the graph, and use normal Excel design procedures. For example, the example graph uses the following data range:

=SERIES(Sheet1!\$A\$59,Sheet1!\$B\$60:\$B\$200,sheet1!\$A\$60:\$A\$200,1)

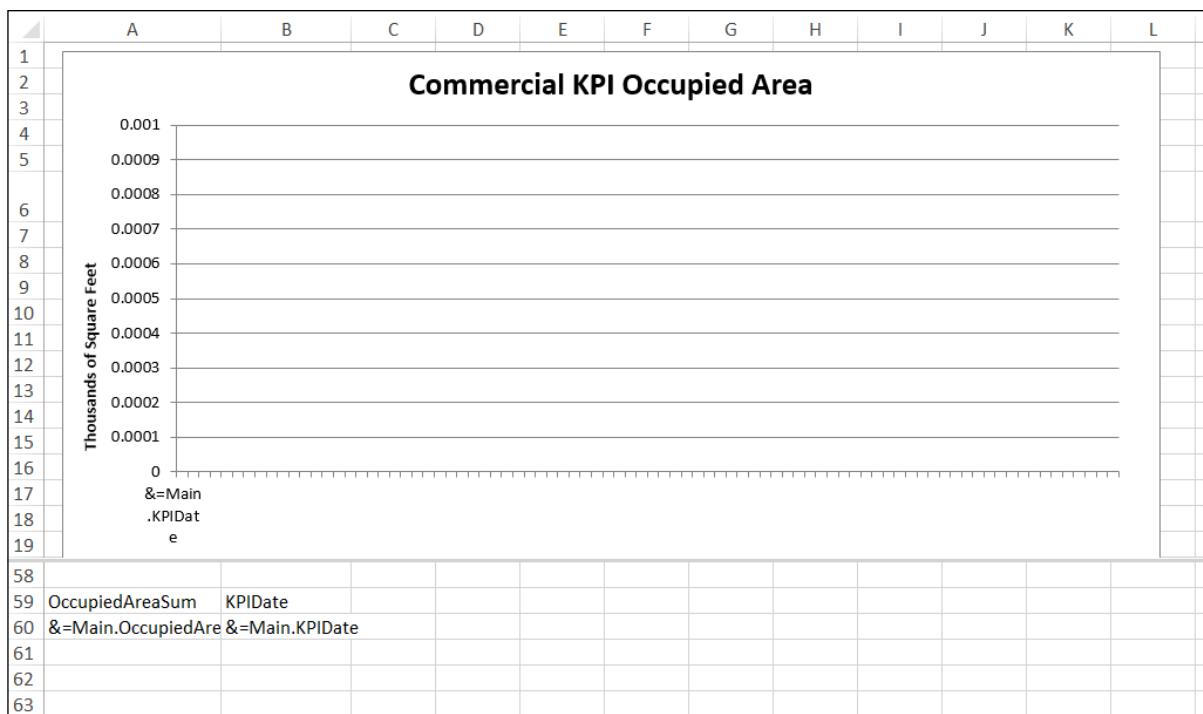
where the syntax for the SERIES formula is as follows:

=SERIES(OptionalSeriesName,OptionalCategoryLabels,RequiredValues,PlotOrder)

- 3** Replace the top line of dummy data with YSR smart markers.

58				
59	OccupiedAreaSum	KPIDate		
60	&=Main.OccupiedAre	&=Main.KP Date		
61	2	February-17		
62	3	March-17		
63	4	April-17		
64	5	May-17		
65	5	June-17		
66	6	July-17		
67	7	August-17		

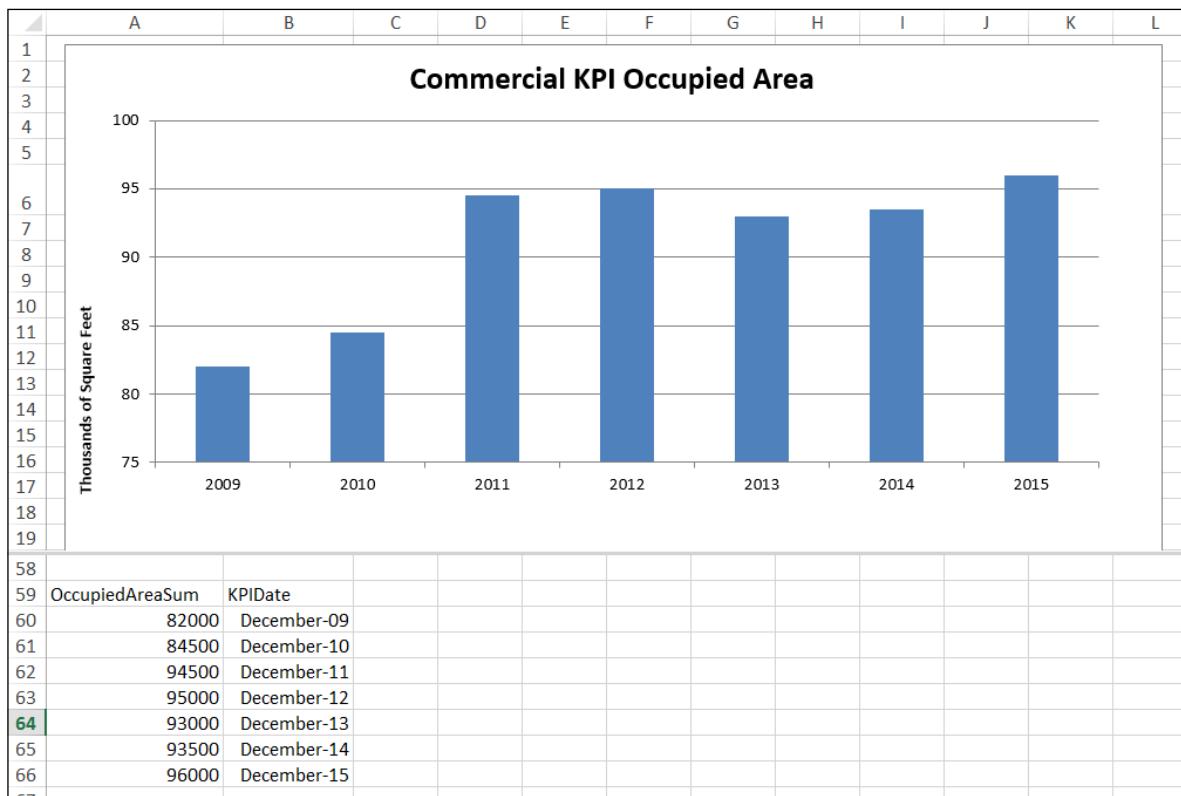
- 4** Delete the contents of the cells containing the dummy data. The graph now appears blank.



Do not delete the cells or rows themselves. The graph must refer to a range of cells sufficient to contain the expanded data generated at run time.

- 5** Save the report template to the Reports path as a .xls file.

- 6 Generate the report. Voyager replaces the smart markers with data and Excel uses the data to render the graph. In the following example, the rendered report returns just seven rows of data. Excel reduces the data ranges accordingly.



If you don't want to see the raw data in the final report, place your dummy data and smart markers on a different worksheet than the graph or hide the rows containing raw data.

### To create a graph with Excel 2013 or 2016

- Add dummy data to your report template.

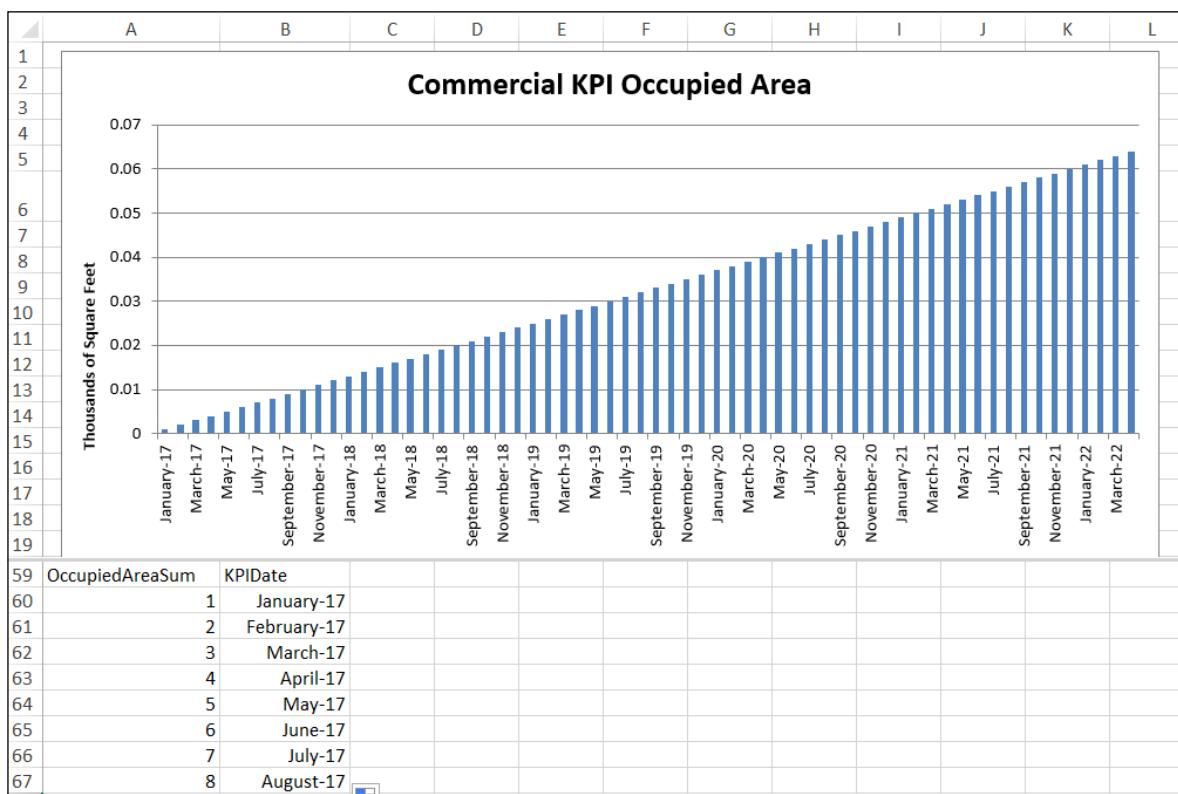
The figure shows a Microsoft Excel spreadsheet with a table of dummy data. The table has two columns: "OccupiedAreaSum" and "KPIDate". The table has 8 rows, with row 64 highlighted in green. The data is as follows:

	OccupiedAreaSum	KPIDate
59	1	January-17
60	2	February-17
61	3	March-17
62	4	April-17
63	5	May-17
64	6	June-17
65	7	July-17
66	8	August-17



Use dummy data that corresponds generally with the data you want to represent (dates for date ranges, numbers for numeric data, and so on).

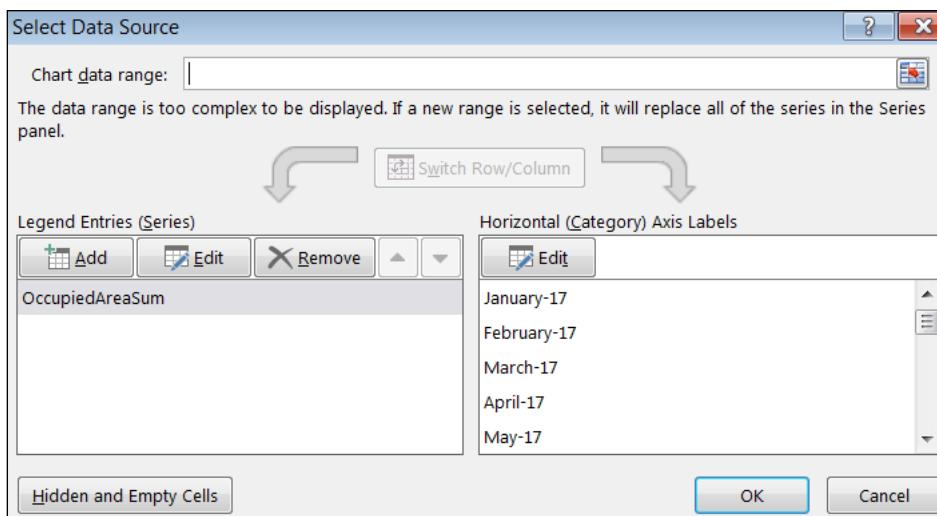
- 2** Add a graph to your template, using the dummy data.



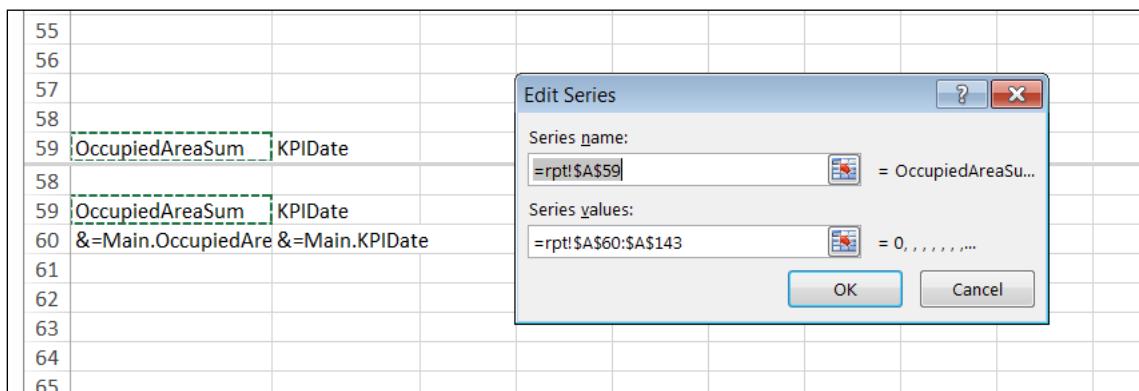
Use the dummy data as the source data for the graph and use normal Excel design procedures.

- 3** Replace the first line of dummy data with smart markers.  
**4** Delete the contents of the cells containing the dummy data. The graph now appears blank.  
**5** If you need to edit the graph data range:

- a Right-click the graph and select **Select Data**. The **Select Data Source** screen appears.



- b To edit data ranges, highlight a data range (on the left, like OccupiedAreaSum in the preceding graphic) and click **Edit**.



- c Edit the fields.

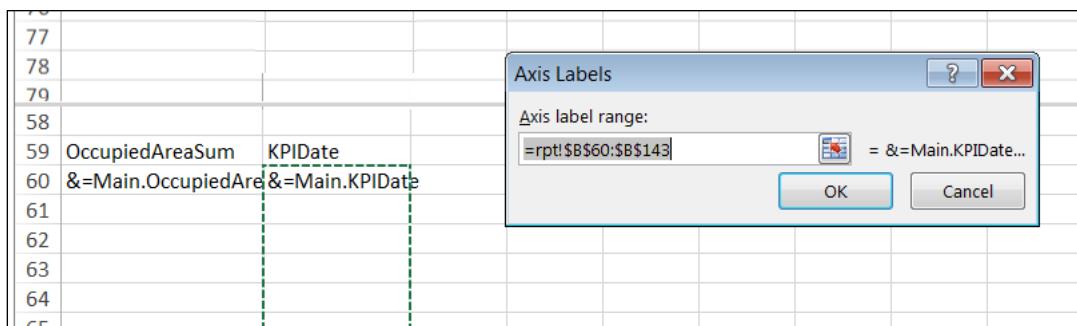
**Series name** The cell that contains the name (the header) for your data. Optional.

**Series values** The data range for the data, to be rendered at run time, you want to represent.

**TIP** The data range must include the cell containing your smart marker, plus sufficient cells for the expanded row data.

- d Click **Ok**. Return to the **Select Data Source** screen.

- e To edit categories (labels), click **Edit** in the **Horizontal (Category) Axis Labels** section.



- f Select a range of cells including your smart marker and sufficient cells for the expanded row data.
- g Click **Ok**.
- 6 Save the worksheet to the Reports path as a .xls (Excel 97-2003) workbook.
- 7 Generate the report.

## Tips and Tricks

### In this section:

Adding Special Formatting to Expanding Row Data .....	233
Date and Number Formatting .....	235
Freeze Panes.....	237
Drill-Down Links .....	238
Displaying Data Horizontally .....	239

This section describes tips and tricks for formatting data with Excel.

### Adding Special Formatting to Expanding Row Data

You can apply special formatting to any smart marker in your Excel report template by using normal formatting commands in Excel. In some cases, however, you may want to add bold, underlining, or other formatting to a line of data within a set of dynamically expanding row data.

For example, some Voyager analytics reports automatically add a subtotal or total row, highlighted in bold, to dynamically expanding data. The Trial Balance report in Financial Analytics, for instance, displays the Total line in bold.

Financial Reports						
Property	comoff01	Account Manager		Denominator		Clear
Book	Accrual	Asset Class		Property		PDF
Account Tree		Building Type		Source		Excel
Report Type	Cash Flow	CFDA		Immediate Source		Display
Period	01/2008 to 12/2010	Grid <input checked="" type="checkbox"/> Freeze Pane <input type="checkbox"/> Decimals <input checked="" type="checkbox"/> Show Property Name <input type="checkbox"/> Suppress Zero <input checked="" type="checkbox"/> Graph <input type="checkbox"/>				
Report Columns	Actual	Summary <input type="checkbox"/> Tree Level 1 <input type="checkbox"/> Show Acct. Code <input checked="" type="checkbox"/> Source <input type="checkbox"/>				
		Period to Date	%	Year to Date	%	
4000-0000	INCOME					
4001-0000	OPERATIONS					
4002-0000	REVENUE					
4003-0000	TENANT RENTS & SERVICES					
4100-0000	OFFICE					
4110-0000	Rent - Office	15,959,778.97	81.37	15,959,778.97	81.37	
4125-0000	Rent - Office Parking	593,920.00	3.03	593,920.00	3.03	
8070-0000	Loan Fees	0.00	500.00	0.00	500.00	
8510-0000	Interest Income	0.00	57,582.73	14,360.73	43,222.00	
8520-0000	Realized Gain/Loss on Sale	0.00	40,750.00	8,150.00	32,600.00	
8550-0000	Advisor Fees	0.00	635.00	635.00	0.00	
	<b>Total</b>	<b>0.00</b>	<b>801,140,395.78</b>	<b>801,139,910.46</b>	<b>485.32</b>	

To apply bold to rows that appear inside a set of dynamically expanding data, use conditional formatting in your Excel report template. With conditional formatting, you can set up a formula that triggers Excel to apply a specific format to a cell, row, or column.

For example, the following graphic shows conditional formatting applied to a cell that contains dynamically expanding data. The formula tells Excel to search for the word **Total** in column A and, where **Total** appears, apply bold and number formatting to one row of data.

The screenshot shows a Microsoft Excel spreadsheet with the following details:

- Formula Bar:** C2 &=[Region].[column1]
- Cells:** Row 1 contains headers: Actual, Budget, Variance. Row 2 contains data: &=Region.acctdesc, &=[Region].[column1], &=[Region].[column2], &=[Region].[column3].
- Conditional Formatting Rule:**
  - Rule Type:** Use a formula to determine which cells to format.
  - Format values where this formula is true:** =FIND("Total",A2)
  - Preview:** Shows a value of \$38,718.00.
- Excel ribbon:** The Home tab is selected. The Conditional Formatting button in the Styles group is highlighted.



This example is only one way to use conditional formatting in Excel. For more information about conditional formatting, see Microsoft Help.



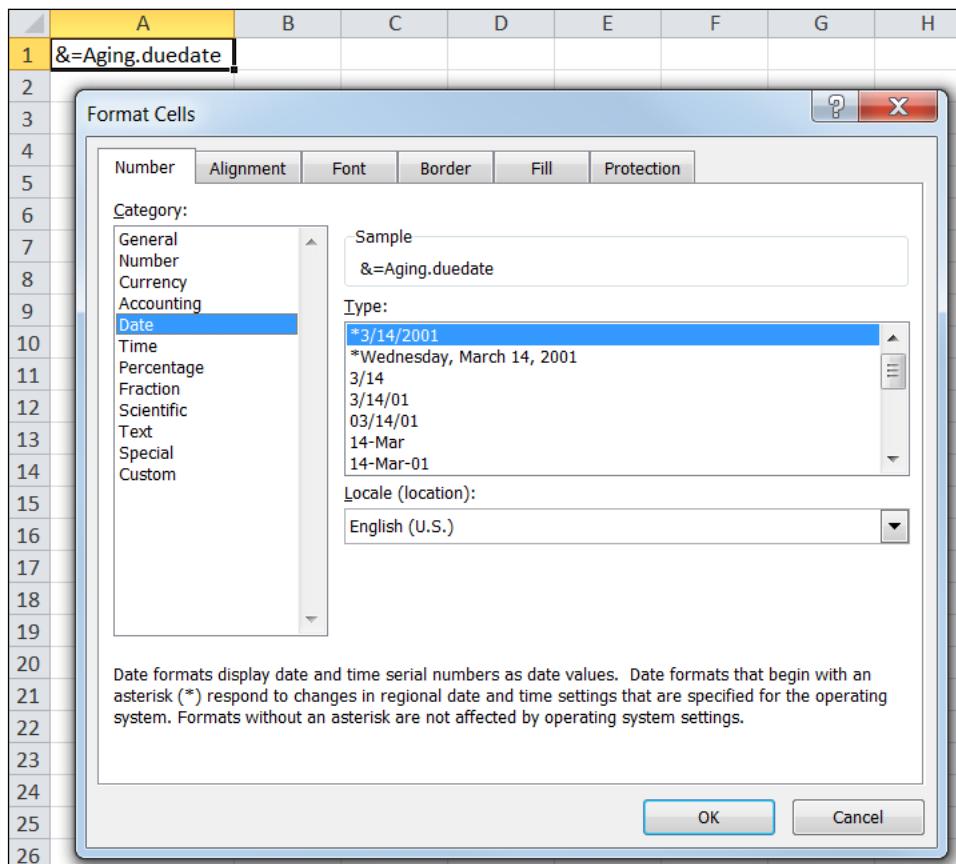
The preceding example uses conditional formatting in Excel 2010. The options available to you depend on the version of Excel you are using.

## Date and Number Formatting

This section describes how to apply date and number formatting.

## Dates

To display dates as per the culture of a secured user in reports published to screen, you must apply date formatting to the cell or cells in your Excel report template. For example, the following graphic shows an Excel spreadsheet set up to display dates in the \*m/dd/yyy format.



## Numbers

To display numbers as per the culture of a secured user in reports published to screen, you must apply number formatting to the cell or cells in your Excel report template. For example, the following graphic shows an Excel spreadsheet set up to display numbers with two decimal places and a minus sign (-) for negative numbers. YSR honors the decimal place formatting in Excel when rendering reports to screen.



YSR also honors percentage formats applied in Excel.

The screenshot shows the 'Format Cells' dialog box in Excel. The 'Number' tab is selected. In the 'Category' dropdown, 'Number' is chosen. The 'Decimal places' input field is set to 2. The 'Sample' field contains the formula '&=Aging.Charge\_Amount'. The 'Negative numbers:' section shows four examples: -1234.10, 1234.10, (1234.10), and (1234.10). A note at the bottom states: 'Number is used for general display of numbers. Currency and Accounting offer specialized formatting for monetary value.'

## Freeze Panes

If you are setting up an Excel template to generate reports on screen, you can use Freeze Pane options in Excel to fix specific rows or columns in place when viewing the report. When you freeze a row or column, Voyager displays scroll bars so that users can scroll through the rest of the report.

To freeze rows or columns, use the **Freeze Panes** item on the **View** menu in Excel. You do not need to alter your smart markers in any way.

The screenshot shows the Microsoft Excel ribbon with the 'View' tab selected. In the 'View' tab's ribbon group, the 'Freeze Panes' button is highlighted. A tooltip for 'Freeze Panes' is displayed, stating: 'Keep rows and columns visible while the rest of the worksheet scrolls (based on current selection)'. Below this, other options like 'Freeze Top Row' and 'Freeze First Column' are listed with their descriptions.

## Drill-Down Links

You can set up Excel templates to contain drill-down links to when publishing reports to screen. To create drill-down links, use the **HyperLink** smart marker syntax and embed references to expanding data objects in the link.

For example, the following string is a **HyperLink** smart marker that creates drill-down links to tenant records, where tenant IDs and tenant names are stored in column E and F of an Excel template:

```
&=&=HyperLink("javascript:DrillDown('../pages/TenantSwitch.aspx?1=1&TenantID='&E{r}&'')",F{r})
```

The syntax of the **HyperLink** smart marker follows:

```
&=&=HyperLink("LinkAddress",LinkCaption)
```



Capitalization is important: you must use **DrillDown**, not Drilldown or any other variation.

You must pass **1=1** as the first parameter. You can then add additional parameters in the following format:  
`&parameter1=X&parameter2=Y&parameter3=Z`

For example, the following graphic shows a column containing a link to the tenant screen:

A	B	C	D	E	F	G
Property	PropID	PropName	Tenant	LeaseID	LeaseName	Units
&=&=HYPERLINK("javascript:Op=&LeaseDet.PropID=&LeaseDet.PropName=&=&=HYPERLINK("javascript:Dr=&=LeaseDet.LeaseID=&=LeaseDet.LeaseName=&=LeaseDet.Units")")						



If you want to retrieve the URL of your webshare for use in hyperlinks, use this token: **#@@WEBSHARENAME#**. For more information about supported tokens, see "Session and User-Related Tokens" on page 52.

## Displaying Data Horizontally

This section provides tips for working with headers, details, and totals when displaying data horizontally.

Displaying data horizontally is simple; you need only add the (horizontal) parameter to your smart markers to cause them to expand laterally instead of vertically. Coordinating header and detail data, however, can be more difficult. The example that follows shows one way that you can link headers to details by dumping data into a hidden worksheet and using the VLOOKUP function to match the detail data to the dynamically expanding column headers.

This illustration demonstrates a number of conventions that you can adapt for other horizontal reports. You can:

- Use {c} inside smart markers to refer to dynamically expanded columns.

For example, &=&={c}2/{c}3~horizontal is a repeating dynamic smart marker. It divides whatever is in row 2 of the current (dynamically expanded) column by whatever is in row 3 of the same column at run time.

- Use the VLOOKUP function to match header and detail data.

For example, suppose that some identifier (like a property code) is arranged horizontally in your main worksheet and vertically on your hidden data dump worksheet. Use the VLOOKUP function in a horizontally expanding, repeating smart marker on the main worksheet. Configure the smart marker to look in the data dump worksheet for data that matches the expanded headers on the main worksheet.

- If the VLOOKUP generates errors like #NA or #REF, wrap the VLOOKUP in an IF statement that uses ISERROR to check for error messages and replace them with a blank string (""). See the example report that follows for an illustration.

### Example horizontal report

Suppose you want to generate a report that displays rent, CAM, and totals, as illustrated here:

	A	B	C	D	E	F	G	H
1								
2		1SADDR1	1SADDR1	1SADDR1	9SADDR1	9SADDR1	9SADDR1	
3		20cq	20cq	20cq	glpm	glpm	glpm	
4		TYPE	LE	YEAR	TYPE	LE	YEAR	
5								
6								
7	RENT		3.00	19.00	6.00	3.00	215.00	6.00
8	CAM		1.50	9.50	3.00	1.50	107.50	3.00
9	TOTAL		4.50	28.50	9.00	4.50	322.50	9.00
10								

To accomplish this, dump your detail data into a hidden worksheet like this:

	A	B	C
1	&=Details.PROPTYPE	&=Details.Rent	&=Details.CAM
2			
3			
4			
5			
6			
7			
8			
9			

At run time, rendered detail data appears (data truncated for space considerations):

	A	B	C
1	TYPE-1SADDR1	3	1.5
2	TYPE-3SADDR1	3	1.5
3	TYPE-4SADDR1	3	1.5
4	TYPE-5SADDR1	3	1.5
5	TYPE-6SADDR1	3	1.5
6	TYPE-7SADDR1	3	1.5
7	TYPE-8SADDR1	3	1.5
8	TYPE-9SADDR1	3	1.5
9	TYPE-14SADDR1	3	1.5
10	LE-1SADDR1	19	9.5
11	LE-3SADDR1	128	64
12	LE-4SADDR1	128	64
13	LE-5SADDR1	212	106
14	LE-6SADDR1	216	108
15	LE-7SADDR1	213	106.5
16	LE-8SADDR1	214	107
17	LE-9SADDR1	215	107.5
18	YEAR-9SADDR1	0	0

Notice that, for purposes of illustration, column A contains concatenated property types and property codes. These are returned separately, in rows 4 and 2 respectively, on the main report worksheet.

	C	D	E	F	G	H
1						
2	1SADDR1	1SADDR1	1SADDR1	9SADDR1	9SADDR1	9SADDR1
3	20cq	20cq	20cq	glpm	glpm	glpm
4	TYPE	LE	YEAR	TYPE	LE	YEAR
5						

Property codes

Property types

To match the detail data with the horizontally expanded header data, therefore, this example report uses dynamically expanding smart markers to concatenate the values in rows 4 and 2 of the expanded columns. The resulting concatenated strings, like TYPE-1SADDR1, match the labels that appear in the data dump worksheet. Therefore they are available for use in a VLOOKUP function as follows:

`&=&=VLOOKUP(CONCATENATE({C}4,"-",{C}2),'DataDump-Report'!A:C,2,TRUE)~horizontal`

and

`&=&=VLOOKUP(CONCATENATE({C}4,"-",{C}2),'DataDump-Report'!A:C,3,TRUE)~horizontal`

These smart markers search for each concatenated string (TYPE-1SADDR1, for example) in the data dump worksheet. They retrieve the corresponding rent values from column 2 and CAM values from column 3, respectively. The 'False' parameter ensures an exact match, rather than an approximate match.

Then, to prevent #N/A and #REF errors from displaying on the main worksheet in case detail data is missing, the example report wraps the VLOOKUP function in an IF statement that checks for errors. The ISERROR function returns 'true' if there is an error, so the IF statement supplies an empty string ("") in case ISERROR = true.

For example (abbreviated for purposes of illustration):

`&=&=IF(ISERROR(VLOOKUP(xyz)), "",(VLOOKUP(xyz))~horizontal)`



The full smart marker for the horizontally expanding rent numbers, without abbreviation, is as follows:

`&=&=IF(ISERROR(VLOOKUP(CONCATENATE({C}4,"-",{C}2),'DataDump-Report'!A:C,2,TRUE)), "",VLOOKUP(CONCATENATE({C}4,"-",{C}2),'DataDump-Report'!A:C,2,TRUE))~(Horizontal)`

Finally, you can sum the rent and CAM numbers with the following smart marker:

`&=&=SUM({c}7+{c}8)~(Horizontal)`

The following graphic shows the report template containing all the smart markers discussed above. (Smart markers are truncated at the edge of the graphic).

	B	C	D	E	F	G	H	I	J	K	L
1											
2		<code>&amp;=Header.PropertyName(Horizontal)</code>									
3		<code>&amp;=Header.PropertyCode(Horizontal)</code>									
4		<code>&amp;=Header.PropertyType(Horizontal)</code>									
5											
6											
7	DATA	<code>&amp;=&amp;=IF(ISERROR(VLOOKUP(CONCATENATE({C}4,"-",{C}2),'DataDump-Report'!A:C,2,TRUE)), "",VLOOKUP(CONCATENATE({C}4,"-",{C}2),'DataDump-Report'!A:C,2,TRUE))~(Horizontal)</code>									
8	DATA	<code>&amp;=&amp;=IF(ISERROR(VLOOKUP(CONCATENATE({C}4,"-",{C}2),'DataDump-Report'!A:C,3,TRUE)), "",VLOOKUP(CONCATENATE({C}4,"-",{C}2),'DataDump-Report'!A:C,3,TRUE))~(Horizontal)</code>									
9	TOTAL	<code>&amp;=&amp;=SUM({c}7+{c}8)~(Horizontal)</code>									
10											

When you generate the report, property codes and types expand dynamically across the template columns and detail data fills in, matched to the right column headers by the VLOOKUP function.

## CHAPTER 8

# Yardi Excel Add-In for YSR

### In this chapter:

Yardi Excel Add-In Overview.....	243
Building YSR Report Templates with the Add-In .....	244
Building YSR Reports with the Add-In (Dump SQL) .....	267

This chapter describes how to use the Yardi Excel Add-In to create a YSR report and design smart markers using the YSR Smart Marker Designer. To locate the Yardi Excel Add-In, search Client Central for **Excel Add**.

The Yardi Excel Add-In provides an alternative to manual design procedures. It is entirely optional; you can design Excel report templates without it, but it makes design procedures much easier. The Yardi Excel Add-In includes the following features:

- A Smart Marker Designer, which you can configure to identify and read from both custom SQL scripts and Voyager analytics.
- A Hyperlink Designer, which you can use to create hyperlinks to Voyager records and screens.
- A report-building function (Dump SQL) that creates a package containing all YSR setup elements.



This chapter describes how to use the Yardi Excel Add-In Release Five. The Yardi Excel Add-In version 4.x supports both YSL.NET (Yardi Spreadsheet Link) and YSR, whereas the Yardi Excel Add-In Release Five supports only YSR. If you develop reports in both YSR and YSL, use the Yardi Excel Add-in 4.x. If you develop reports in YSR only, use the Yardi Excel Add-in Release Five.

For information about using the Yardi Excel Add-In version 4.x or earlier with YSR, see earlier versions of the *Yardi Spreadsheet Reporting User's Guide*.

For information about how to use the Yardi Excel Add-In version 4.x or earlier with YSL.NET, see the *Yardi Spreadsheet Link User's Guide*.

## Yardi Excel Add-In Overview

The Yardi Excel Add-In helps you design Excel templates suitable for use in YSR. The Add-In is optional, since you can manually type all elements of a YSR Excel template. However, the Add-In can significantly assist your design efforts with tools such as the YSR Smart Marker designer (which provides information about field names exposed by Voyager analytics or scripts), the YSR Hyperlink Designer, auditing tools, and additional documentation material.

### Advantages of the Yardi Excel Add-In

With the Yardi Excel Add-In, you can:

- Create smart markers automatically with the YSR Smart Marker Designer.
- Create hyperlinks with the YSR Hyperlink Designer.
- View Voyager analytics filter usage documentation.
- Build a Word merge token file, for use with Word report templates.
- Retrieve the field aliases used in Voyager analytics (the recommended method for building smart markers for any report that retrieves data with Voyager analytics).
- Retrieve the field aliases used in your pool of YSR Yardi Script files.
- Dump SQL: create a .pkg file containing all the design specifications for your YSR report.
- Check your Excel template for errors and list the conditional formats imposed.

### Limitations of the Yardi Excel Add-In

The report-building (Dump SQL) function of The Yardi Excel Add-In currently:

- Creates YSR reports containing one sub-report only (one .txt SQL script file and one report template).
- Is for use with Excel templates only (not Word).
- Requires at least one SQL select statement (the Add-In does not write selects for you, and it does not currently build reports that use Voyager analytics for retrieving data).

### Yardi Excel Add-In Workflow

The following is an example of a typical workflow.

- 1 Set up the Yardi Excel Add-In.
  - a Download and install the Yardi Excel Add-In.
  - b Enable the YSR Smart Marker Designer to appear on right-click.

**2** Create an Excel template with smart markers.

**a** Select your data source or sources.

**b** Build smart markers.

**c** Create hyper-links.

**d** Check smart markers for errors.

**3** Build a YSR report.

**a** Select your data source.

**b** Create custom filters fields for use when generating the report.

**c** Configure top-level select statement.

**d** Dump SQL.

**e** Load SQL package into Voyager using Voyager Workstation Administration.

**f** Save your Excel template and SQL script to the Reports path.

**4** Generate your report.

## Building YSR Report Templates with the Add-In

**In this section:**

Downloading and Installing the Yardi Excel Add-In.....	245
Downloading YSR Design Helper .....	248
Setting the Default YSR Design Helper Path.....	249
Enabling the YSR Smart Marker Designer .....	250
Selecting Data Sources.....	251
Creating Smart Markers with the YSR Smart Marker Designer.....	254
Creating YSR Hyperlinks with the YSR Hyperlink Designer .....	258
Checking Your Excel Template for Errors .....	262
Checking Your Excel Template for Conditional Format Rules .....	263
Building Merge Token Files for Word Templates .....	264
Viewing YSR Analytic Filter Usage Documentation .....	265

This section describes how to use the Yardi Excel Add-In to prepare report templates for YSR reports.

## Downloading and Installing the Yardi Excel Add-In

To use the Yardi Excel Add-In, first download the Add-In from Client Central. To find the Add-In, search Client Central for **Excel Add**. Select the version that you want to use:

Module	Submodule	Description
Reporting > Correspondence, & Analytics	YSL (Yardi Spreadsheet Link)	Yardi Excel Add In YSL.Net the XLA
Reporting > Correspondence, & Analytics	YSR (Yardi Spreadsheet Reporting)	Yardi Excel Add In Release Five

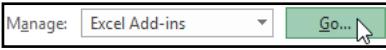
If you develop reports in YSR only, select the Yardi Excel Add-in Release Five. Next, install the Add-in.

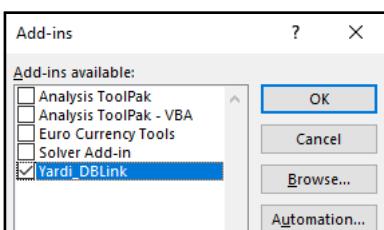


This procedure and the procedures that follow describe how to install and use the Yardi Excel Add-In in Office 365. Your screen may appear differently depending on your version of Excel and the other add-ins that you have installed (if any).

### To install the Yardi Excel add-in

- 1 From the Excel top menu, click **File**.
- 2 From the side menu, click **Options**. The **Excel Options** screen appears.
- 3 From the side menu, click **Add-ins**.
- 4 In the **Manage** field, select **Excel Add-ins**.

- 5  Click **Go**. The **Add-ins** screen appears.



- 6 If **Yardi\_DBLink** appears in the list, select the **Yardi\_DBLink** check box.
- 7 If **Yardi\_DBLink** does not appear in the list:
  - a Click **Browse** and locate the **Yardi\_DBLink.xla** file.



You can download the latest version of **Yardi\_DBLink.xla** from Client Central. Although you can save the file to any folder on your computer (excluding the desktop), you should save it to a subfolder in your user's **Documents** folder. When you need to update the file, you can replace the new copy with the old one.

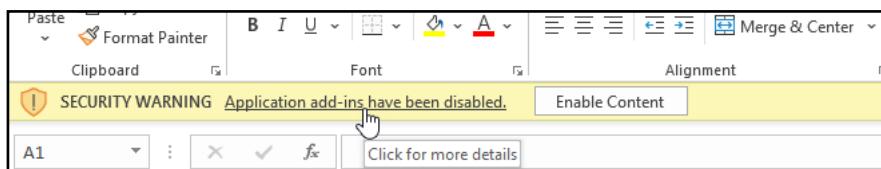
You can always review where the Yardi Excel Add-In has been previously installed by selecting **Add-ins > Display Yardi Excel Add-In path** from the top menu. Closing Excel and replacing the existing **Yardi\_DBLink.xla** file at that location with a new copy will immediately update Excel using the same established security settings.



If you save the file to a network location, including Microsoft OneDrive, you must select the **Allow Trusted Locations on my network (not recommended)** check box in the **Trusted Locations** section on the **Trust Center** screen in Excel. Ensure that enabling this setting is in compliance with your company's IT policy. For information about opening the **Trust Center** screen, see below.

By saving the file to a network location, you can centrally update it for all machines that access it from there. However, the add-in will not be available if the network location becomes offline (except if the location supports local synchronization, such as with OneDrive).

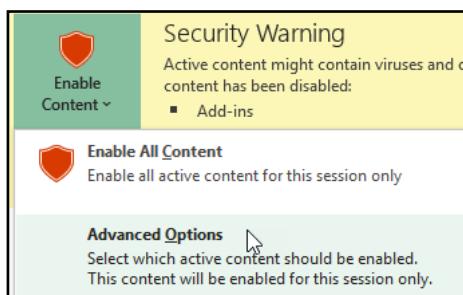
- b** Click **OK**. The **Add-Ins** screen appears.
  - c** Select the **Yardi\_DBLink** check box.
- 8** Open a blank workbook. A **Security Warning** appears below the ribbon.



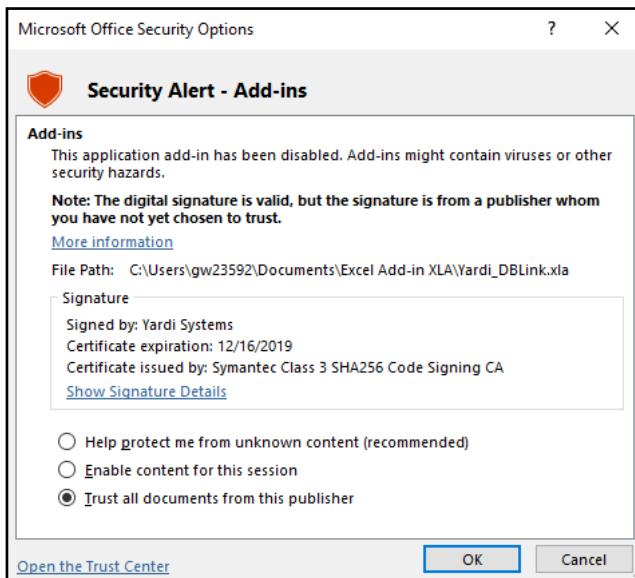
If you click **Enable Content**, you can use the add-in for this session only. You will need to click **Enable Content** each time you open Excel. Instead, you should grant Yardi Systems access as a trusted publisher by following the remaining steps in this procedure.

If Excel has detected security issues with multiple items, the displayed messaging might slightly vary from what appears in this procedure.

- 9** Click **Application add-ins have been disabled**. The **Info** screen appears, with the **Security Warning** section.



**10** Click **Enable Content**, and then select **Advanced Options**. The **Microsoft Security Options** screen appears.

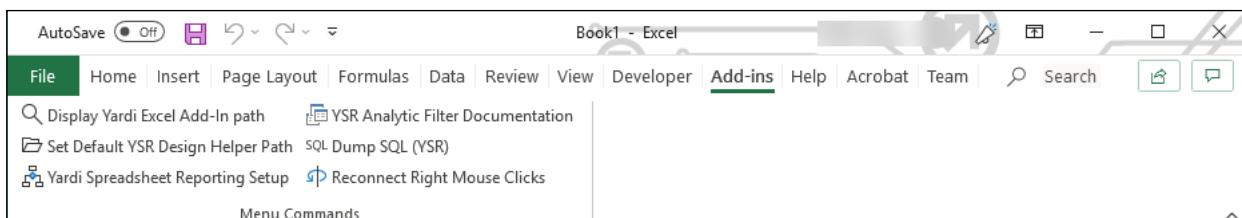


**11** Select **Trust all documents from this publisher**.



Before you click **OK**, ensure that your network administrator or your corporate IT policies allow you to grant such trust.

**12** Click **OK**. New menu commands for the **Yardi Excel Add-in** appear on the **Add-ins** tab.



**13** If you want to confirm that Yardi Systems is now a trusted publisher:

- From the Excel top menu, click **File**.
- From the side menu, click **Options**. The **Excel Options** screen appears.
- From the side menu, click **Trust Center**, then click **Trust Center Settings**. The **Trust Center** screen appears.
- From the side menu, click **Trusted Publishers**. The following screen appears.

Trusted Publishers	
Trusted Locations	Issued To
Trusted Documents	Yardi Systems
Trusted Add-in Catalogs	Issued By
	Symantec Class 3 SHA256 Code Signing CA
	Expiration Date
	12/16/2019
	Microsoft Code Signing PCA
	7/26/2019

## Downloading YSR Design Helper

By downloading the YSRDesignHelper.xml file from a Voyager database, you can limit the analytics interfaces that you can select when viewing YSR Analytic Filter Usage documentation to those available in the database. This ensures that you view only the information that is relevant to the database when designing Excel templates with the add-in.

### To download YSR Design Helper

- 1 From the **YSR Admin** side menu, select **Setup > YSR Administration**. The **YSR Administration** screen appears.



For information about the YSR Admin side menu, see "Adding the YSR Admin Menu" on page 162.

- 2 On the **Export Excel Add-In Design Time XML** tab, click **Export Yardi Excel Add-In Design Helper XML**.
- 3 Follow the prompts to save the YSRDesignHelper.xml file.



Save the file to your documents folder and use the dialogue in the Yardi Excel Add-in to add this .xml file to the add-in. For more information, see below.



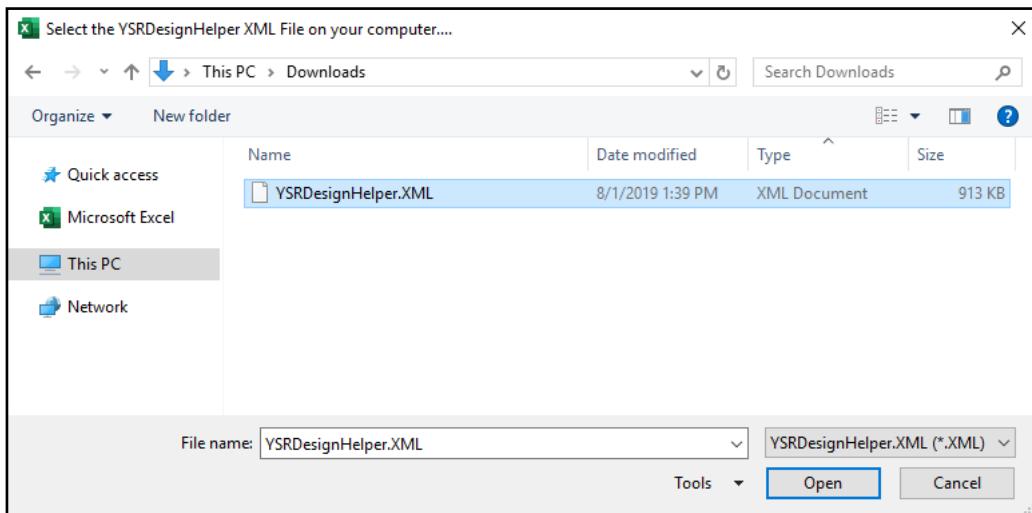
YSRDesignHelper.xml is not compatible with YSLDesignHelper.xml.

## Setting the Default YSR Design Helper Path

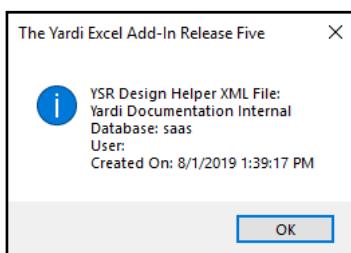
After you've downloaded the YSRDesignHelper.xml file, you must set the default path to it from the Yardi Excel Add-in.

### To set the default YSR Design Helper path

- From the Excel top menu, select **Add-Ins > Set Default YSR Design Helper Path**.



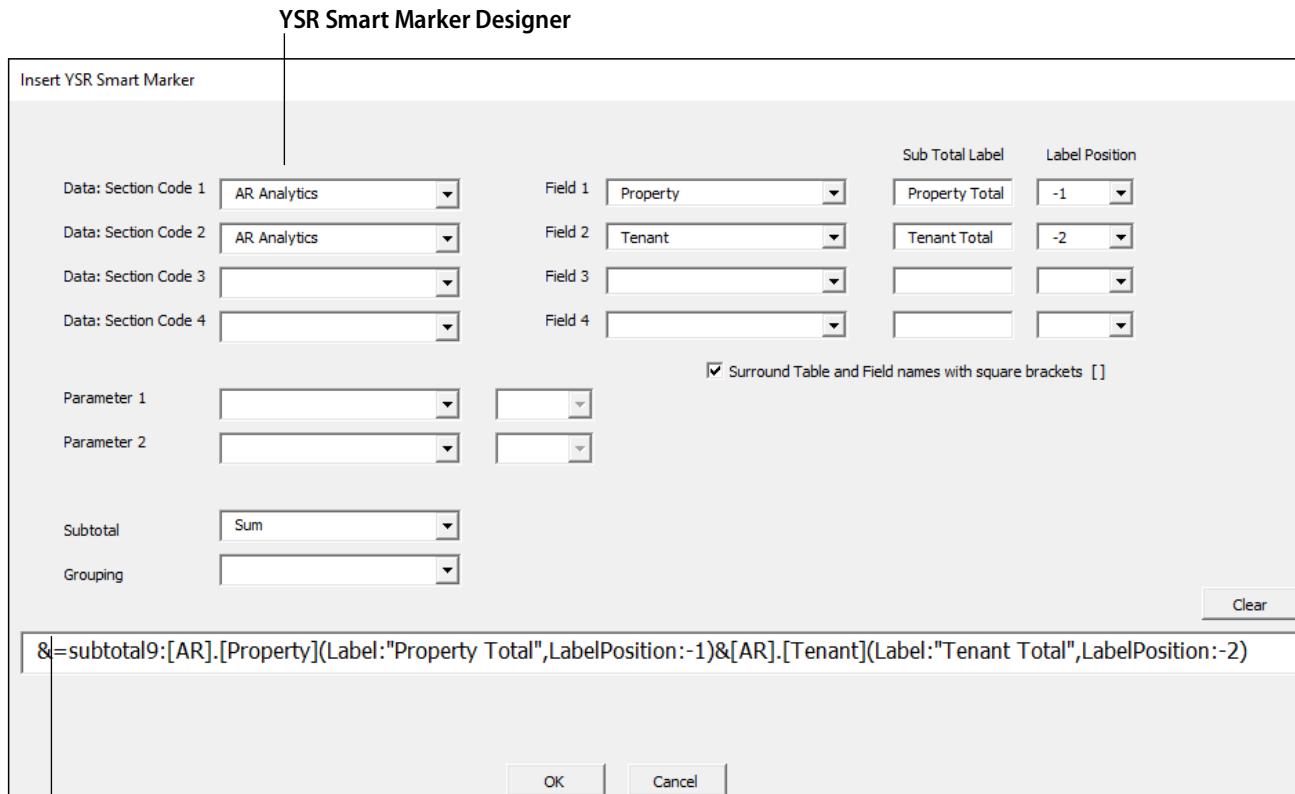
- Select the YSRDesignHelper.xml file, and then click **Open**. A confirmation message appears with information about the source database and when the file was downloaded.



## Enabling the YSR Smart Marker Designer

To take advantage of the Yardi Excel Add-In's capacity to format smart markers automatically, enable the YSR Smart Marker Designer.

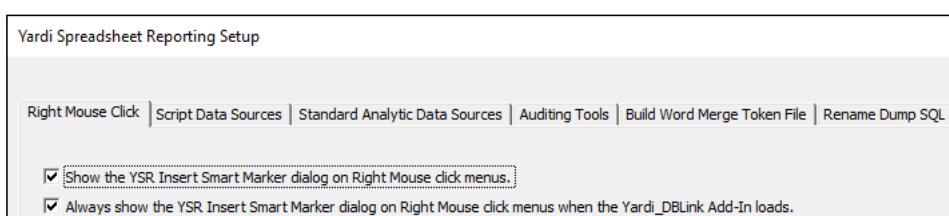
After you enable the right mouse click option and select a data source or sources, you can right-click in any cell in the worksheet and choose **YSR Smart Marker**. The **Insert YSR Smart Marker** screen appears, where you can build smart markers. The documentation refers to this screen as the YSR Smart Marker Designer.



Smart marker in process of being built

### To enable the YSR Smart Marker Designer on right mouse-click

- From the Excel top menu, select **Add-Ins > Yardi Spreadsheet Reporting Setup**. The **Yardi Spreadsheet Reporting Setup** screen appears.
- Select the **Right Mouse Click** tab.



- Select the check box in the top row to add the YSR Smart Marker Designer to your right-click menu options.

- 4** If you want the YSR Smart Marker Designer to appear on right-click menu options always, select the check box in the second row.
- 5** Click **Ok**.

## Selecting Data Sources

When you specify data sources, the select section names and field aliases used in data retrieval become available to the YSR Smart Marker Designer and to Dump SQL (report-building) operations.

### Data Sources

There are two types of data sources, custom SQL script files and Voyager analytics.

- If you are using the Yardi Excel Add-In to design smart markers, select as many data sources as your YSR report contains (both custom SQL scripts and Voyager analytics).
- If you are using the Yardi Excel Add-In to create an entire report (using **Dump SQL**), select just one custom SQL script file as your data source.



The Yardi Excel Add-In currently builds reports containing one sub-report only (one .txt SQL script file and one .xls report template). The script file can contain multiple named selects, however.

### Data Source Refresh

If you make changes to a custom SQL script in an external text editor and save the changes, you can refresh your data source so that the YSR Smart Marker Designer reflects the changes in your script. When you refresh your data source, any new or edited select section names or field aliases become available to the YSR Smart Marker Designer. For more information, see “To refresh a script data source” on page 253.

### Yardi Excel Add-In Settings and Configuration

All settings and configuration data related to the Yardi Excel Add-In are specific to your computer. If you change computers, you must reconfigure the Yardi Excel Add-In.

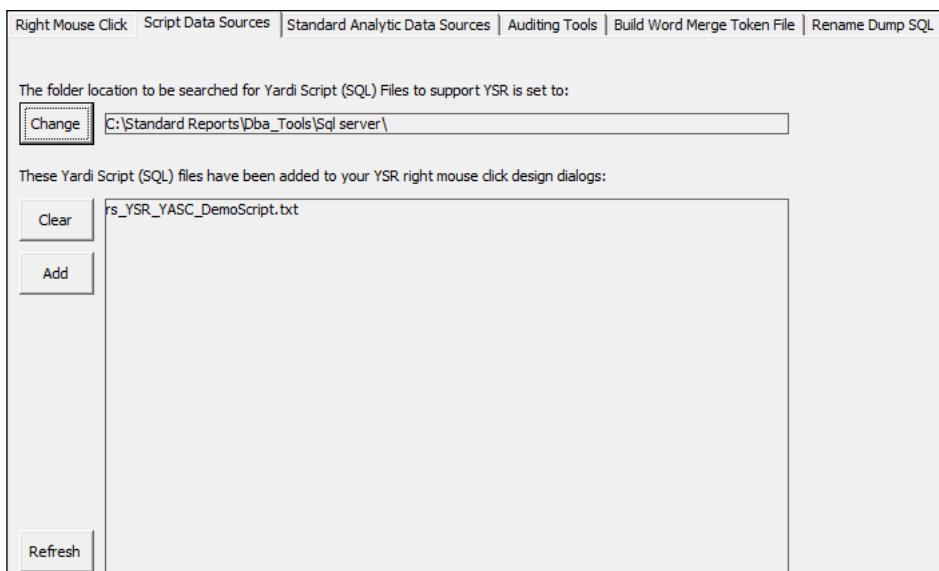
#### This section includes the following procedures:

To select a custom SQL script as a data source.....	251
To refresh a script data source .....	253
To select a Voyager analytics report as a data source.....	253

#### To select a custom SQL script as a data source

- 1 From the Excel top menu, select **Add-Ins > Yardi Spreadsheet Reporting Setup**. The **Yardi Spreadsheet Reporting Setup** screen appears.

**2 Select the Script Data Sources tab.**



**3 Click Change.** A dialog box appears.

**4 Navigate to the script files that accompany your report.**

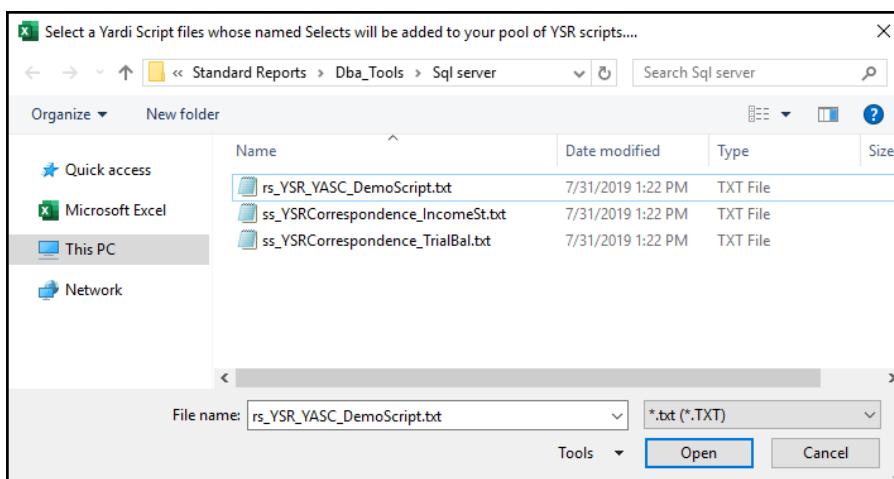
**5 Select a file and click Open.** The filename of the script appears in the main box of the **Script Data Sources** tab, and the file path of the file appears in the **Change** field.



The file that you select here is the first custom SQL script that becomes available to the Smart Marker Designer. You can use just one script, or you can add multiple script files.

**6 If you want to add other script files as data sources:**

**a Click Add.** A dialog box appears.



**b Navigate to the script files that accompany your report.**

- c Select a file and click **Open**. The file appears in the main section of the **Script Data Sources** tab of the **Yardi Spreadsheet Reporting Setup** screen.
  - d Repeat this process to add additional script files.
- 7 Click **Ok**. The select section names and field aliases used in the scripts become available to the YSR Smart Marker Designer and to report-building operations.

### To refresh a script data source



When you refresh a script data source, the select section names and field aliases used in the script become available to the YSR Smart Marker Designer. Refreshing your script does not have any effect on Dump SQL (report-building) operations. You must re-build your report package in order to reflect changes to your script.

- 1 From the Excel top menu, select **Add-Ins > Yardi Spreadsheet Reporting Setup**. The **Yardi Spreadsheet Reporting Setup** screen appears.
- 2 Select the **Script Data Sources** tab.
- 3 Click **Refresh**.
- 4 Click **OK**. Changes to section names and field aliases take effect in the YSR Smart Marker Designer and in Dump SQL (report-building) operations.

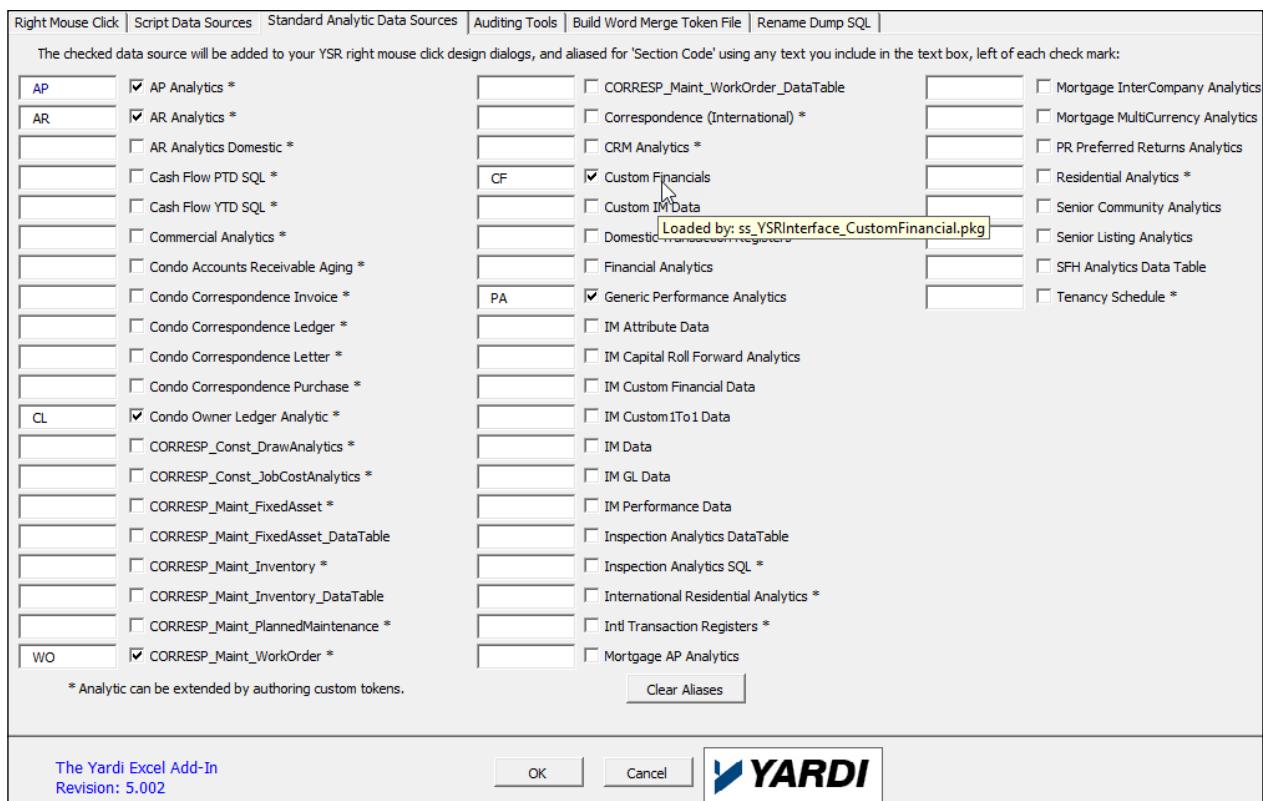
### To select a Voyager analytics report as a data source



The Yardi Excel Add-In does not currently support Voyager analytics for report-building (Dump SQL). Select a Voyager analytics data source for building smart markers only.

- 1 From the Excel top menu, select **Add-Ins > Yardi Spreadsheet Reporting Setup**. The **Yardi Spreadsheet Reporting Setup** screen appears.

## 2 Select the Analytic Standard Data Sources tab.



If you want to view the package name of a Voyager analytics report, place the cursor on the name of the report.

- 3 Select the check box corresponding to the Voyager analytics report used in your YSR report.
- 4 Click **OK**. The select section names and field aliases used in the analytics report become available to the YSR Smart Marker Designer.

## Creating Smart Markers with the YSR Smart Marker Designer

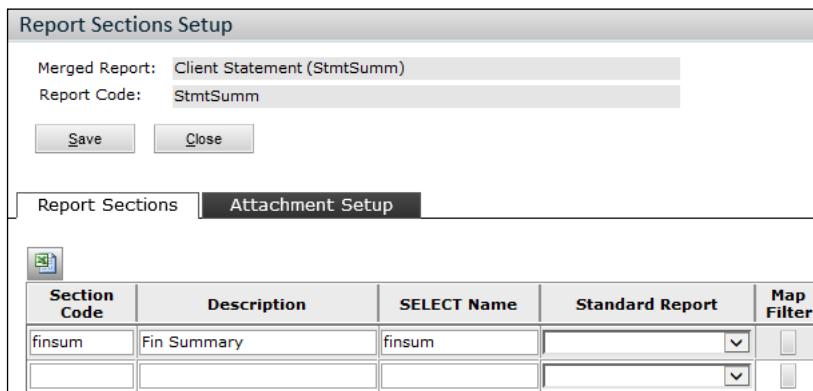
After you select the data sources used in your YSR report, the select section names and field aliases used in the data source become available to the YSR Smart Marker Designer. With the designer you can format smart markers, assign parameters, and set up subtotals, groups, and labels.



The YSR Smart Marker Designer cannot create formula or image smart markers. For more information about how to design these smart markers manually, see "Excel Formulas and References" on page 206 and "Images" on page 220.

## Select section names and section codes

When you use the YSR Smart Marker Designer to create a smart marker, the Designer identifies the select section names in your data sources and uses them as the section codes in your smart markers. Therefore, when you complete the **Report Section Setup** screen, you must use the full select section name as the section code, rather than using a shorthand code. The following graphic displays a report section in which the section code is the same as the select section name.



If you must use a different section code, you can still use the YSR Smart Marker Designer, but you have to use Excel's global replace feature to replace the select section name (provided by the YSR Smart Marker Designer) with the section code (as per your report design).

YSR distinguishes between select section names and section codes.

- Select section names are the names of Select statements, as they appear in custom SQL Scripts.

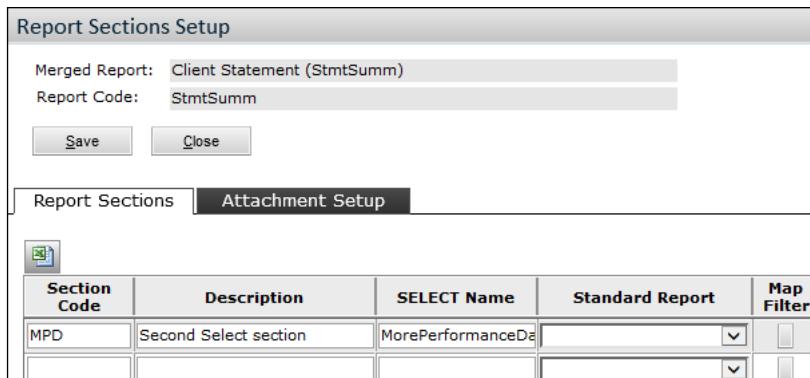
```
//Select MorePerformanceData
SELECT
    p.scode          Propcode
    ,ASC16.TOTWATERUSE TotalWaterUsage
    ,ASC16.TOTGASUSE TotalGasUsage
    ,ASC16.LASTYEAR GAS LASTYearGasUsage
FROM
Property p
Left Join ASCAP16 ASC16 on ASC16.HPROP = p.hmy
WHERE 1 = 1
AND ASC16.Dtdate = dbo.perf_eop('#AsofMonth#', 'm')
#Conditions#
//End Select
```

- Section codes are codes for section names. Section codes appear in smart markers.

	A	B
1	&=[MorePerformanceData].[TotalGasUsage]	
2		
3		

Section code

When manually adding smart markers, you might use section codes instead of full select section names as a matter of convenience. For example, you might use MPD as a section code for MorePerformanceData. Whether or not you use different section names and codes, however, you must map your codes to your names on the **Report Sections Setup** screen.



The distinction between section codes and names is important because the YSR Smart Marker Designer uses the full select section name as the section code. You must therefore complete the **Report Sections Setup** screen using the same string for the section code and the select section name.

When you use the YSR Smart Marker Designer, the Designer first identifies the section names in your data sources and then uses them as section codes in your smart markers.

#### To create a smart marker with the YSR Smart Marker Designer

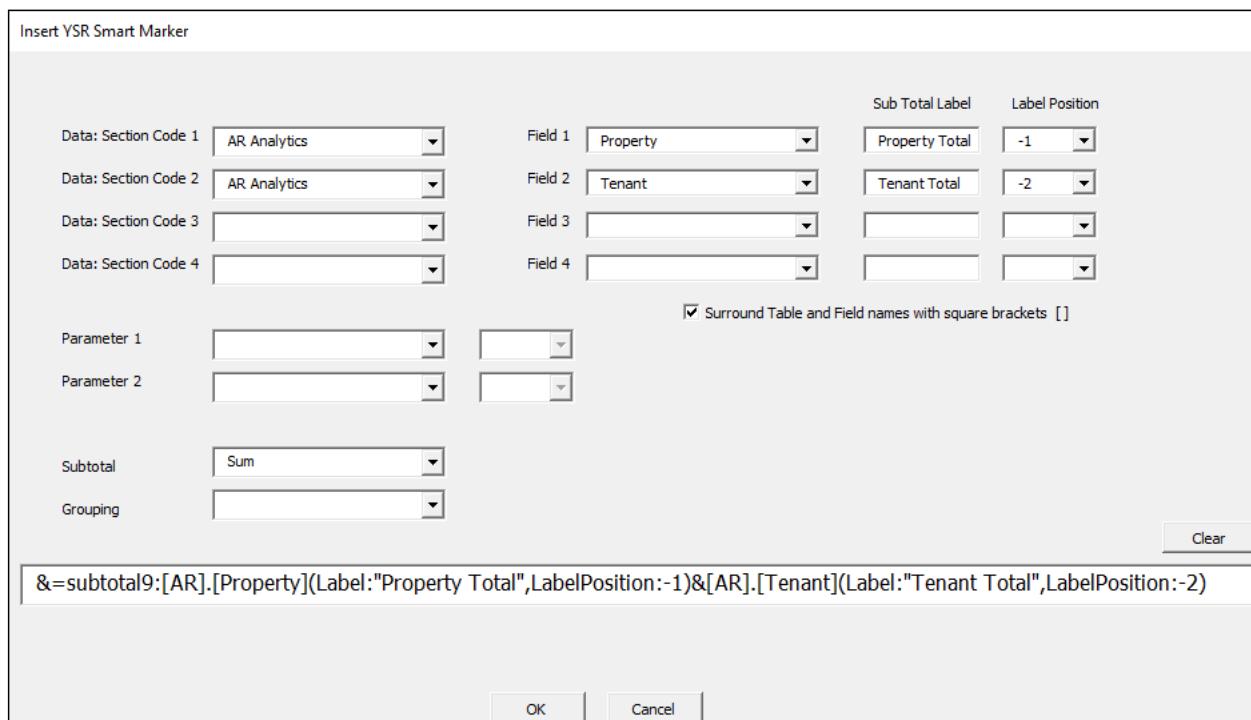
- 1 In your Excel template, right-click in any worksheet cell. A set of right-click menu options appears.
- 2 Select **YSR Smart Marker**. The **Insert YSR Smart Marker** screen appears.
- 3 Complete the fields. For more information, see “Insert YSR Smart Marker Screen Reference” on page 257.



As you make selections, the smart marker appears in the long horizontal field at the bottom of the screen.

- 4 Click **Ok**. The YSR Excel Add-In inserts the smart marker into the Excel worksheet.

## Insert YSR Smart Marker Screen Reference



### Data: Section Code 1-4

The name of the select section that retrieves the data the smart marker displays.

The YSR Smart Marker Designer provides a list of the select section names that appear in the data sources you identify when configuring the Designer. Choose the select section name that you want to use as the section code in your smart marker.

For more information about selecting data sources, see “Selecting Data Sources” on page 251.

For more information about the difference between section names and codes, see “Select section names and section codes” on page 255.

**NOTE** Rows two through four become editable only when you make a selection in the **Subtotal** field. Complete more than one row only when you want to perform Excel calculations on multiple groups of data. For example, you can use two rows to display a subtotal of charges by property and a subtotal of charges by tenant. For more information about calculating subtotal for data groups, see “Subtotaling Multiple Data Groups” on page 213.

### Field 1-4

The field alias of the data represented by the smart marker.

The options available in this field depend on your selection in the corresponding **Data: Section Code** field. Once you make a selection, the YSR Smart Marker Designer displays the field aliases from that select section in the **Field** drop-down list. Choose the field alias you want to use in your smart marker.

**NOTE** Rows two through four become editable only when you make a selection in the **Subtotal** field. Complete more than one row only when you want to perform Excel calculations on multiple groups of data. For example, you can use two rows to display a subtotal of charges by property and a subtotal of charges by tenant. For more information about calculating subtotal for data groups, see “Subtotaling Multiple Data Groups” on page 213.

### Subtotal Label

For use with subtotal smart markers. The label of the subtotal.

<b>Label Position</b>	The position of the subtotal label with respect to the smart marker. -1 indicates one cell to the left, -2 indicates two cells to the left; 1 indicates one cell to the right, 2 indicates two cells to the right, and so on.
<b>Surround Table and Field Names with square brackets [ ]</b>	<p>When selected, the YSR Smart Marker Designer surrounds the section codes and field names with brackets. By using square brackets, you can include embedded spaces in your section codes and field names.</p> <p><b>TIP</b> You must select this option if your section codes and field names contain embedded spaces.</p> <p><b>TIP</b> You can use square brackets for section codes and field names without embedded spaces, as well. Turn square brackets off only if you are sure you will never have embedded spaces in section codes or field names.</p>
<b>Parameter 1-2</b>	<p>Formatting options. For more information about each option, see "Smart Marker Parameters" on page 204.</p> <p><b>TIP</b> Some parameters require additional specifications. For example, the <i>skip</i> parameter, which skips rows, requires you to state how many rows you want to skip. Use the additional field to the right of the <b>Parameter 1</b> field to make additional specifications. For example, select 2 to skip two rows.</p>
<b>Subtotal</b>	<p>The Excel function performed by the smart marker (optional).</p> <p><b>TIP</b> By making a selection in this field, you can create a subtotal smart marker. Subtotal smart markers can perform all of the statistical aggregation functions offered in Microsoft Excel, the most common of which is Sum. Scroll through the list to find the function you want to perform, and the Designer formats the smart marker with the correct numbering (for example, Sum corresponds to subtotal9 in Microsoft Excel). For more information about subtotals, see "Subtotaling a Single Data Group" on page 212.</p> <p><b>NOTE</b> When you make a selection in this field, rows two through four of the YSR Smart Marker Designer become editable. You can use the additional rows to create subtotals for multiple data groups. For example, you can display a subtotal of charges by property and a subtotal of charges by tenant. For more information about calculating subtotal for data groups, see "Subtotaling Multiple Data Groups" on page 213.</p>
<b>Grouping</b>	<p>The method YSR uses to cluster repeated data items. For examples and more information, see "Data Groups" on page 211.</p>

## Creating YSR Hyperlinks with the YSR Hyperlink Designer

With the YSR Hyperlink Designer, you can create hyperlinks to Voyager objects (records) or specific Voyager pages. You can define up to eight parameters for each hyperlink, based on either hard-coded values or values derived from other cells in the Excel template.

Each YSR hyperlink occupies a single cell in an Excel template, and may use the following columns:

- 1 One column to display the text to be shown by the hyperlink on the report (for example, a tenant's name). Typically, this is the column to the left of the hyperlink.
- 2 One column to contain the hyperlink. This column does not appear on the report.
- 3 One to eight columns to define parameters for the hyperlink. These columns appear on the report. You can also manually define parameters with hard-coded values.

Suppose you want to create a hyperlink that opens a tenant record using the tenant's ID and the tenant's unit's ID as parameters. You want to display the hyperlink as the name of the tenant.

In a new row on the Excel template, you enter the following values:

Item	Value	Description
A Display	&=Tenant.TenantName	Displays the hyperlink as the tenant's name.
B Hyperlink	(Blank)	You will insert a YSR hyperlink in this cell.
C Parameter 1	&=Tenant.TenantId	The first parameter is the tenant's id number
D Parameter 2	&=Tenant.TenantUnitId	The second parameter is the tenant's unit's ID number.

The following graphic shows what the Excel template might look like after you enter these values:

	A	B	C	D	E
1	Display	Hyperlink	Parameter 1	Parameter 2	Parameter 3
2	TenantName		TenantId	UnitId	
3	&=Tenant.TenantName		&=Tenant.TenantId	&=Tenant.TenantUnitId	

By right-clicking cell D3 and selecting **YSR Hyperlink**, you can create the following hyperlink with the YSR Hyperlink Designer:

```
&=&=HYPERLINK ("javascript:DrillDown (' /pages/CommonTenant.aspx?
1=1&TenantId=&C{r}&"&UnitId=&D{r}&" ') ",A{r})
```

You should follow this general layout convention for every hyperlink that you create. By following this convention, you anticipate the default behavior of the YSR Hyperlink Designer.



Do not create YSR hyperlinks that direct to Voyager YSR financial reports. The YSR Hyperlink Designer financial object types. Auditing requirements for financial reports require that the reports be completely centralized within the Voyager financial analytics engine. For information about using custom financial analytics, see *Voyager Custom Financial Analytics Reporting User's Guide*.

### To create a YSR Hyperlink with the YSR Hyperlink Designer

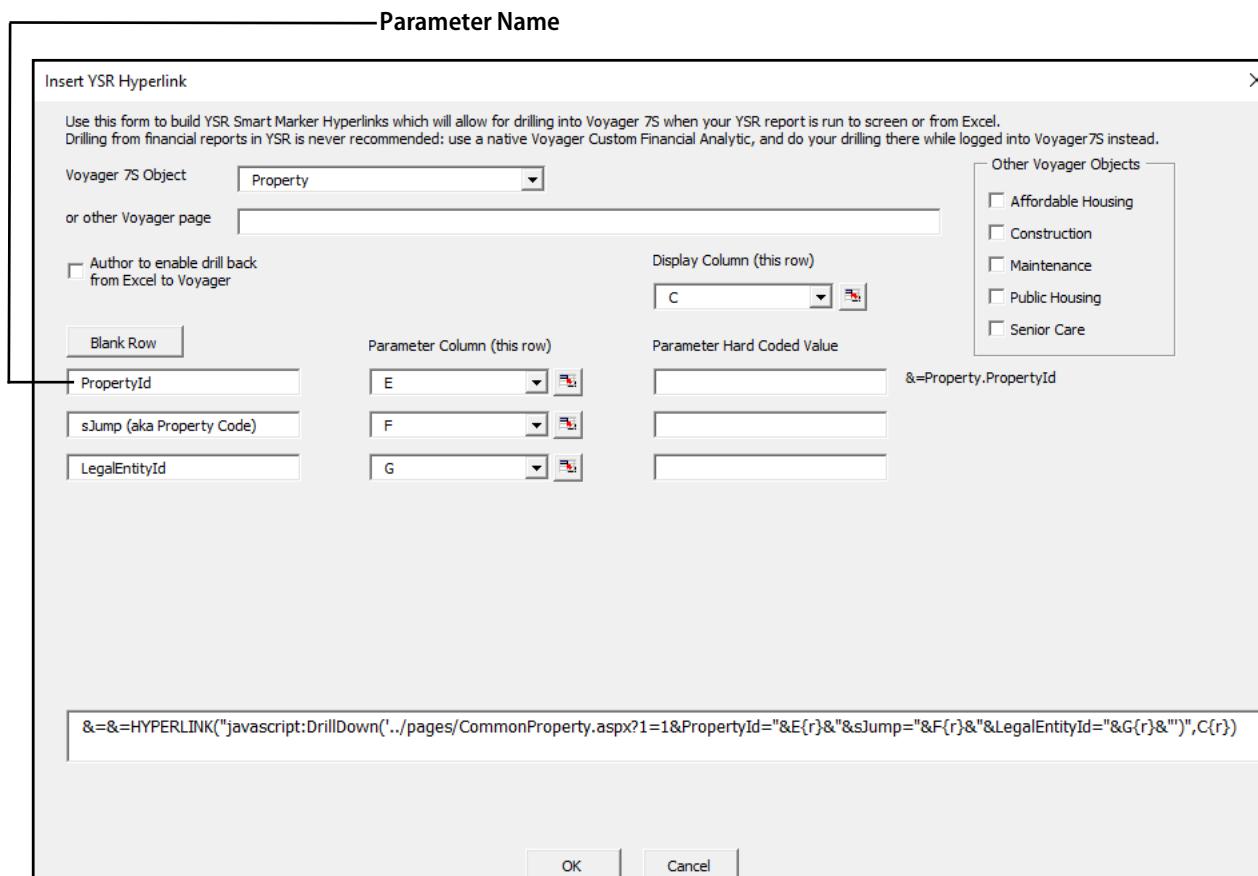
- 1 In your Excel template, right-click the worksheet cell where you want to insert the hyperlink. A set of right-click menu options appears.
- 2 Select **YSR Hyperlink**. The **Insert YSR Hyperlink** screen appears.
- 3 Complete the fields. For more information, see “Insert YSR Hyperlink Screen Reference” on page 260.



As you make selections, the hyperlink appears in the long horizontal field at the bottom of the screen.

- 4 Click **OK**. The YSR Excel Add-In inserts the hyperlink into the Excel worksheet.

## Insert YSR Hyperlink Screen Reference



<b>Voyager 7S Object</b>	If you want to link to an object (record) type, such as properties or tenants, select the object type. If you want to link directly to a Voyager page, leave this field blank and complete the <b>Or Other Voyager Page</b> field below.
<b>Other Voyager Objects</b>	Select the Voyager modules with the object types that you want to appear in the <b>Voyager 7S Object</b> field.
<b>Or Other Voyager Page</b>	<b>NOTE</b> The YSR Hyperlink Designer remembers your module selections when you create additional hyperlinks in the future.
<b>Author to enable drill back from Excel to Voyager</b>	If you want to link directly to a Voyager page, type the file name of the page. Ensure that you include the .aspx extension (for example, Page.aspx). If you want to link to an object type, leave this field blank and complete the <b>Voyager 7S Object</b> field above.

#@@WEBSHARENAME# <b>Column (this row)</b>	(Appears only if you select the <b>Author to enable drill back from Excel to Voyager</b> check box.) Column in the same row that contains the webshare address, which is required for hyperlinks opened from Excel. Insert #@@WEBSHARENAME# into the column. This token returns the active webshare address.  <b>TIP</b> You can also define the webshare address by including the following select statement in the report:  SELECT '#@@WEBSHARENAME#' as ( <i>webshare address</i> )  However, you should use the token instead, as it dynamically returns the webshare address so that the report can be generated on multiple databases.
<b>Display Column (this row)</b>	Column in the same row that you want to use to display the hyperlink. The column should contain a value to determine the display text. By default, this is the column to the left of the column where you inserted the hyperlink.
<b>Parameter Names</b>	If you selected an option in the <b>Voyager 7S Object</b> field, the YSR Hyperlink Designer automatically populates each row on the screen with supported parameters for the object type.  If you manually entered a page name in the <b>Or Other Voyager Page</b> field instead, you must manually complete this field for each row on the screen.  <b>TIP</b> If you want to add a blank row for a custom parameter, click the <b>Blank Row</b> button. This button appears only if you selected an option in the <b>Voyager 7S Object</b> field. It does not appear if you manually entered a page name in the <b>Or Other Voyager Page</b> field.  <b>TIP</b> You can define the parameter directly from this field with an equals sign, followed by the parameter value (for example Action=FILTER). If you do this, the <b>Parameter Column (this row)</b> and <b>Parameter Hard Coded Value</b> fields become unavailable.  <b>TIP</b> You can insert explanatory notes about the parameter with parentheses (), prefaced with or without a space. The YSR Hyperlink Designer ignores these notes when generating the hyperlink.  <b>NOTE</b> Although object types can support more than one parameter, you will often only need to use one to successfully create a hyperlink. You can clear the rows for the other parameters. For example, if you selected <b>Property</b> , the YSR Hyperlink Designer automatically populates rows for the <b>PropertyId</b> , <b>sJump</b> (property code), and <b>LegalEntityId</b> parameters. You can successfully link to a property using only the <b>PropertyId</b> parameter, so you can clear the other rows. You might need to experiment with each object type to identify which parameters are optimal to use.
<b>Parameter Column (this row)</b>	If you want to define the parameter by referring to a cell in the same row but in a different column, select the column.  If you want to manually define the parameter, complete the <b>Parameter Hard Coded Value</b> field instead.  <b>TIP</b> If the column in the same row contains a value, it appears to the right of the <b>Parameter Hard Coded Value</b> field.  <b>NOTE</b> If you clear the value in this column and in the <b>Parameter Hard Coded Value</b> column, the hyperlink will not use the parameter.
<b>Parameter Hard Coded Value</b>	Value of the parameter.  If you want to define the parameter by referring to a cell in the same row but in a different column, complete the <b>Parameter Column (this row)</b> field instead.  <b>NOTE</b> If you clear the value in this column and in the <b>Parameter Column (this row)</b> column, the hyperlink will not use the parameter.

## Checking Your Excel Template for Errors

You can use the Yardi Excel Add-In to check for errors in your smart markers.

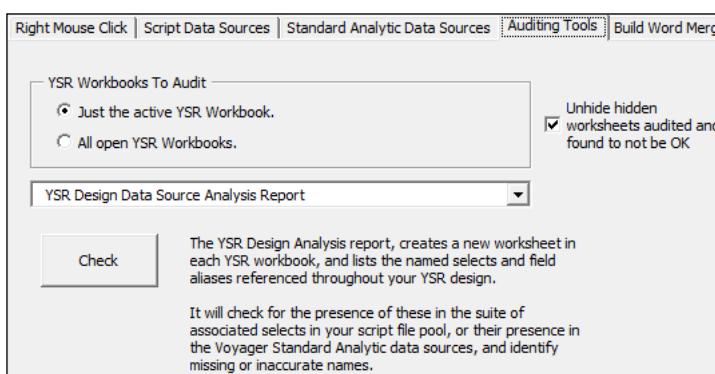
When you perform an audit, the Yardi Excel Add-In examines your smart markers to make sure that their section codes and field aliases match the section codes and field aliases used in your data sources.

The Yardi Excel Add-In displays the audit results in your Excel template, in a new worksheet titled **YSR Design Analysis**. For example, the following graphic shows the results of an audit in which there are five correctly formatted smart markers and one problematic smart marker. You can link directly from the **YSR Design Analysis** to a specific smart marker by clicking on the link in the **Location** column (column E).

	B	D	E	F
1	Status	Worksheet	Location	Named Sele
OK Analytic or Named Select.				
2 Field Name not found.	ASCAP17	<a href="#">\$E\$2</a>		ASCAP17
3 OK	ASCAP12	<a href="#">\$A\$2</a>		ASCAP12
4 OK	ASCAP12	<a href="#">\$AA\$2</a>		ASCAP12
5 OK	ASCAP12	<a href="#">\$AB\$2</a>		ASCAP12
6 OK	ASCAP12	<a href="#">\$AC\$2</a>		ASCAP12
7 OK	ASCAP12	<a href="#">\$AD\$2</a>		ASCAP12

### To check an Excel template for errors

- From the Excel top menu, select **Add-Ins > Yardi Spreadsheet Reporting Setup**. The **Yardi Spreadsheet Reporting Setup** screen appears.
- Select the **Auditing Tools** tab.



- Complete the **YSR Workbooks To Audit** section.



To ensure that you can drill-down directly from the audit results to any erroneous smart markers, select the **Unhide hidden worksheets audited and found to not be OK** check box. You cannot drill-down from the audit results to hidden worksheets.

- 4 Select the **YSR Design Data Source Analysis** report in the field below.
- 5 Click **Check**. The YSR Excel Add-In checks your Excel template and displays results on a new worksheet titled **YSR Design Analysis**.



The results worksheet displays the status of all the audited smart markers. Smart markers with problems appear at the top. You can click each location link in column E to navigate directly to specific smart markers.

## Checking Your Excel Template for Conditional Format Rules

You can use the Yardi Excel Add-In to check for any cells with conditional format rules. Conditional format rules are especially important for financial reports, which require row-by-row formating based on returned values.

The Yardi Excel Add-In displays the audit results in your Excel template, in a new worksheet titled **YSR Conditional Format**.

A	B	C	D	E	F	G	H	I	
1	Workbook	Worksheet	Range Address	Formula1	Type	AppliesTo.Address	Borders.LineStyle	Font.Bold	Font.FontStyle
2	YSR_MyReport.xlsx	Sheet1	\$E\$7	=\${Y6}<>0	(2) Expression	\$E\$6,\$E\$7		TRUE	Bold
3	YSR_MyReport.xlsx	Sheet1	\$G\$7	=\${X6}<>0	(2) Expression	\$G\$6,\$G\$7		TRUE	Bold
4	YSR_MyReport.xlsx	Sheet1	\$I\$6	=\${Y6}<>0	(2) Expression	\$I\$6:\$I\$7,\$M\$6:\$N\$7		TRUE	Bold
5	YSR_MyReport.xlsx	Sheet1	\$I\$6	=\${X6}<>0	(2) Expression	\$I\$6:\$I\$7,\$M\$6:\$N\$7		TRUE	Bold
6	YSR_MyReport.xlsx	Sheet1	\$J\$6	=\${Z6}<>0	(2) Expression	\$J\$6:\$J\$7,\$M\$6:\$O\$7		TRUE	Bold
7	YSR_MyReport.xlsx	Sheet1	\$I\$7	=\${Y6}<>0	(2) Expression	\$I\$6:\$I\$7,\$M\$6:\$N\$7		TRUE	Bold
8	YSR_MyReport.xlsx	Sheet1	\$I\$7	=\${X6}<>0	(2) Expression	\$I\$6:\$I\$7,\$M\$6:\$N\$7		TRUE	Bold
9	YSR_MyReport.xlsx	Sheet1	\$J\$7	=\${Z6}<>0	(2) Expression	\$J\$6:\$J\$7,\$M\$6:\$O\$7		TRUE	Bold
10	YSR_MyReport.xlsx	Sheet1	\$M\$6	=\${V6}<>0	(2) Expression	\$I\$6:\$I\$7,\$M\$6:\$N\$7		TRUE	Bold

### To check an Excel template for conditional format rules

- 1 From the Excel top menu, select **Add-Ins > Yardi Spreadsheet Reporting Setup**. The **Yardi Spreadsheet Reporting Setup** screen appears.
- 2 Select the **Auditing Tools** tab.

Yardi Spreadsheet Reporting Setup

Right Mouse Click | Script Data Sources | Standard Analytic Data Sources | **Auditing Tools** | Build Word Merge

YSR Workbooks To Audit

Just the active YSR Workbook.  
 All open YSR Workbooks.

Unhide hidden worksheets audited and found to not be OK

Conditional Format Report

**Check**

The Conditional Format report, creates a new worksheet in each YSR workbook, and lists all the cells and ranges that have Conditional Format rules established within the Excel design.

**3 Complete the YSR Workbooks To Audit section.**



To ensure that you can drill-down directly from the audit results to any cells with conditional format rules, select the **Unhide hidden worksheets audited and found to not be OK** check box. You cannot drill-down from the audit results to hidden worksheets.

**4 Select the Conditional Format Report report in the field below.**

**5 Click Check.** The YSR Excel Add-In checks your Excel template and displays results on a new worksheet titled **YSR Conditional Format**.

## Building Merge Token Files for Word Templates

To prepare Word report templates, you must use Word merge fields to stand in for Voyager data. With some versions of Word, you can insert merge fields manually. With Word 2013, however, you must link to a data source before you can insert merge fields.

To satisfy this requirement, you must provide Word with a text file containing a list of the tokens used in your YSR report. For an example, see “Working with Word 2013” on page 99.

You can prepare a token file manually, or you can use the Yardi Excel Add-In to build the file automatically.

### To build merge token file for Word templates

- 1 Open Excel.**
- 2 From the Excel top menu, select Add-Ins > Yardi Spreadsheet Reporting Setup.** The **Yardi Spreadsheet Reporting Setup** screen appears.
- 3 Select the Build Word Merge Token File tab.**



**4 Make a selection in the Token Sorting section.**

<b>Alphabetically Within Sections</b>	When you view the token list in Word, tokens are sorted by section. Within each section, tokens are listed alphabetically.
<b>Alphabetically Globally</b>	When you view the token list in Word, all tokens are listed alphabetically. They are not organized by section.

5 Click **Build**. The Excel Add-In builds a token file. Note the file location.

6 Click **Ok**.

The next step is to link the token file to your Word report template so that you can insert merge fields. For more information, see "Working with Word 2013" on page 99.

## Viewing YSR Analytic Filter Usage Documentation

When mapping filters for analytics report sections, you must hard-code any filter values that you want to remain constant. You should be familiar with the acceptable values for each filter that a report uses.

Using the Yardi Excel Add-in, you can view exhaustive lists of the values that have been used (to date) as constant values for Voyager analytics data sources in YSR. You can also view the percentage of reports that use each filter.

The data listed in the Yardi Excel Add-in comes from the entire suite of YSR reports authored for inclusion in Voyager Plug-ins, as well as those created historically in response to YSR Custom Programming Requests by Yardi Systems.



The Yardi Excel Add-in displays some information that is available in Appendix D, "Constant Values for Analytics Data Source Filter Mapping." The Yardi Excel Add-in also displays additional usage analysis data, and so it is the recommended information source that you should use when mapping filters.

### To view YSR Analytic Filter Usage documentation

- 1 From the Excel top menu, select **Add-Ins > YSR Analytic Filter Documentation**. The **YSR Analytic Filter Usage Documentation** screen appears.
- 2 In the **YSR Standard Analytic Interface** field, select the analytics interface that you want to view documentation for. The documentation appears below. For more information, see "YSR Analytic Filter Usage Documentation Screen Reference" on page 266.

## YSR Analytic Filter Usage Documentation Screen Reference

% of YSR reports populating each filter item	Collection of values or suggested source	Data Type	% of YSR reports populating each filter item	Collection of values or suggested source	Data Type
100 ReportType	16 'ReportType' have been set		8 BookId	use a ysiBook Lookup	
86 StartDate	use a date filter		8 CustomerIds	use a ysi Customer lookup	
47 SummaryType	6 'SummaryType' have been set		8 EndDate	use a date filter	
45 MainQuery	2 'MainQuery' have been set		8 CtrlNoFrom		
45 PropertyList	use a ysi Property List lookup		8 CtrlNoTo		
33 IsDetail	4 'IsDetail' have been set		8 CustomerStatus		
33 PropertyIds	use a ysi Property or Property List lookup		4 LeaseStatus	3 'LeaseStatus' have been set	
29 IsMarketRent	1 'IsMarketRent' has been set		4 TopXFact	2 'TopXFact' have been set	
16 AttributeValue			4 AccountIds	use a ysiAccount lookup	
16 BuildingList			4 VendorIds	use a ysi Vendor lookup	
16 ChargeIds			4 TenantId	use a ysi Tenant lookup	
16 ChargeList			4 ContactIds	use a ysi Contact lookup	
16 ShowFutureActiveLe			4 ContactRole		
16 ShowPendingAmend			4 CountryIds		
16 SpecialtyLeasesEna			4 CustomerCode		
12 PropertyCode	2 'PropertyCode' have been set		4 CustomerTypeIds		
12 PeriodType	3 'PeriodType' have been set		4 LeaseType		
12 PropertyOwner	use a ysi Owner lookup		4 TopCount		
12 LeaseName			2 AttributeCode	2 'AttributeCode' have been set	
12 TenantList					

Limit to YSR Standard Analytic Interfaces in database 'saas' from YSR Design Helper XML extracted on 8/1/2019 1:39:17 PM

The Yardi Excel Add-In Release Five for YSR  
(c)2019 Yardi Systems Inc.

**YARDI**

<b>% of YSR records populating each tier item</b>	Percentage of reports that use the filter. Includes YSR reports authored for inclusion in Voyager Plug-ins, as well as those created historically in response to YSR Custom Programming Requests by Yardi Systems. Filters are sorted by this percentage, from most to least frequently used.
<b>Collection of values or suggested source</b>	<p>If this field appears as a list, the first list item indicates how many constant values have been set for the filter (for example: "3 'PeriodType' have been set.") The following list items indicate the values that were set for the filter (for example: <b>Annually</b>, <b>Monthly</b>, and <b>Quarterly</b>.)</p> <p>If this field appears as a text field, there are no constant values for the filter, but the text indicates the suggested source to use (for example: "use a date filter.").</p> <p>If this field appears as a blank field, there are no constant values for the filter and there are no suggested sources.</p> <p><b>NOTE</b> You can edit this field, regardless of how it appears, with your own notes and annotations. Any changes you make will not be saved after you close the <b>YSR Standard Analytic Filter Usage Documentation</b> screen.</p>
<b>Data Type</b>	<p>Displays the data type of the filter (for example: Boolean, Int32, String).</p> <p>To display data in this column, you must download and set up the YSRDesignHelper.xml file. For more information, see below.</p>
<b>Limit to YSR Standard Analytic Interfaces in database '[name]' from YSR Design Helper XML extracted on [timestamp]</b>	<p>Limits the options that appear in the YSR Standard Analytic Interface field to analytics interfaces that are available in the database associated with the YSRDesignHelper.xml file that you downloaded from Voyager and set up in the Yardi Excel add-in.</p> <p>If you have not yet downloaded or set up the file, the field name changes to <b>UNAVAILABLE - no YSR Design Helper XML has been loaded</b>. For more information, see "Downloading YSR Design Helper" on page 248 and "Setting the Default YSR Design Helper Path" on page 249.</p>

# Building YSR Reports with the Add-In (Dump SQL)

## In this section:

Preparing Script Files For Use With the Yardi Excel Add-In.....	268
Adding a Top-Level Select Statement.....	269
Building the //Filter Section .....	270
Filter Mapping and the Value1 Parameter.....	272
Val1 Parameter Conventions .....	272
Building Report Packages (Dump SQL) .....	273

This section describes how to build a YSR report using the Yardi Excel Add-In.

## Prerequisites

In order to build a report using the Add-In, you need:

- A .txt script file containing one or more select statements.
- The file path for the location of the script.
- Access to Voyager Workstation Administration.

## Setup steps

- 1 Prepare a data source (a .txt script file containing one or more select statements).
- 2 Prepare an Excel report template.
- 3 Build a YSR report.
  - Select your script file as the data source for your report.
  - Name your YSR report.
  - Specify whether or not you are using a top-level select statement.
  - Build and map your filters (custom filter fields).
  - Create a .pkg file (Dump SQL).
- 4 Load the package through Voyager Workstation Administration.
- 5 Upload your Excel report template and SQL script to the Reports path.
- 6 Generate your report.

## Preparing Script Files For Use With the Yardi Excel Add-In

When using the Yardi Excel Add-In to build a YSR report, you must prepare a .txt file containing at least one select statement. For example, the following text shows a basic YSR script with one select statement named *Main*:

```
//SELECT Main
Select
k.dOccupiedArea      OccupiedAreaSum
,k.dtDate            KPIDate
,p.sCode             PropCode
FROM ASCAP_CommKPI k
INNER JOIN Property p on p.hmy = k.hprop
WHERE 1=1 #Condition1#
And k.dtDate > DATEADD(year,-7,(Select max(k2.dtDate) from ASCAP_CommKPI k2))
ORDER BY k.dtDate
//END SELECT
```

YSR scripts need to contain select statements only; they do not need to obey all Yardi scripting conventions. For more information, see “YSR Scripting Conventions” on page 51.

### Tokens and Filtration

When preparing a script file for use with the Yardi Excel Add-In, there are two ways to apply filtration to //Select statements. You can use named tokens like #Property# in the WHERE clause, or you can use numbered tokens (#Conditionn#) in the WHERE clause.

If you use named tokens, the Yardi Excel Add-In replaces the named tokens with numbered tokens when you build the SQL package. For a limited set of named tokens, the Yardi Excel Add-In builds an accompanying filter field and supplies the corresponding Value1 parameter. The Value1 parameter (for example, **p.hmy in (#Property#)**) is appended to the WHERE clause at run time.

For more information about filters and the Value1 parameter, see “Filter Mapping and the Value1 Parameter” on page 272 and “Val1 Parameter Conventions” on page 272.



Because the Yardi Excel Add-In replaces named tokens with numbered tokens, you must add the //Filter section to your script file if you want to reuse the script file in future YSR reports. (If you leave the script file without adding a //Filter section, it contains numbered tokens (#Conditionn#) only. Numbered tokens do not provide the Yardi Excel Add-In any information for building a corresponding filter field.) You can add the //Filter section to your script file by clicking **Commit Changes to Script** on the **Build Your Script Filter** screen.

For more information about building filters with the Yardi Excel Add-In, see “Building the //Filter Section” on page 270.

## Adding a Top-Level Select Statement

The top-level select statement is optional for any report that contains just one Excel report template. All other YSR reports require a top-level select statement.

For background information about the top-level select statement, see “Report Design Considerations” on page 23.

### How to add a top-level select statement

There are two ways to add a top-level select statement when using the Yardi Excel Add-In:

- Add an unnamed select statement to your script file. For example:

```
//SELECT
Select p.sCode propCode, p.sAddr1 PropAddress FROM Property p
Where 1=1 #Conditions#
//END SELECT
```



If you include the top-level select statement in your script, leave it unnamed. Then, select **My Script has an unnamed //Select** on the **Dump SQL** tab of the **Yardi Spreadsheet Reporting Dump SQL** screen. Enter the key column in the **Key Column** field of the **Yardi Spreadsheet Reporting Dump SQL** screen.

- Enter the top-level select statement on the **Dump SQL** tab of the **Yardi Spreadsheet Reporting Dump SQL** screen and complete the **Key Column** field.

### Tokens and Filtration in the Top-Level Select Statement

The top-level select statement may include all filter tokens by referencing #Conditions# in an evolving WHERE clause (for example: WHERE 1=1 #Conditions#).

Alternatively, the top-level select statement may include selective, named tokens such as #Property#. When you click **Dump SQL** to build your report package, the Yardi Excel Add-In recreates the named tokens in the numbered (#ConditionN#) format, where N corresponds to the row number of the filter on the **Build Your Script Filter** screen.

For more information about filtration, see “Building the //Filter Section” on page 270 and “Filter Mapping and the Value1 Parameter” on page 272.

## Building the //Filter Section

In the Yardi Excel Add-In, the word *filters* refers broadly to all elements of filtration. When you use the Yardi Excel Add-In to build filters, you are:

- Creating the custom filter fields that users see on the report generation screen:

- Creating the //Filter section of the SQL in your report package, which you may or may not save to your SQL script file. The next graphic shows an example of a single filter field as coded into a //Filter section. The //Filter section has ten columns separated by commas, eight of which must be defined for use with YSR.

```
//FILTER
C,          I,          Property,          Property,          ,3,
p.hMy      in (#Property#),,Y,          Y,          ,
//END FILTER
```

- Mapping the //Filter section to the //Select sections in your SQL script file.

The following graphic shows the //Filter section and custom filter fields in the process of being built in the Yardi Excel Add-In:

Element Type		Data Type	Name	Caption	List OR Lookup Type	Val1	Mandatory?	Multi Select?	Parent
1	L - List	T - Tex	Property	Property	ysiPropertyLookup (3)	p.hmy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	L - List	T - Tex	Books	Books	Cash^Accrual^Consolidation^Elimin	b.book	<input type="checkbox"/>	<input type="checkbox"/>	
3							<input type="checkbox"/>	<input type="checkbox"/>	
4							<input type="checkbox"/>	<input type="checkbox"/>	

Automatically Rename Named Tokens to Numbered Conditions in Script File on Dump SQL (recommended on first use)



You can build up to nine filter elements with the Yardi Excel Add-In. If your YSR report requires more than nine filters fields, you must build them in Voyager. For more information, see "YSR Report Filter Definition" on page 62.

When you use the Yardi Excel Add-In to build a filter, you can (optionally) save the build as a full, syntactically correct //Filter section in your SQL script by clicking **Commit Changes to Script**. Then the Yardi Excel Add-In edits and saves the textual content of the script file, adding a //Filter section like the following:

```
//FILTER
L,           T,           Property,       Property,      ,3,
L,           T,           Books,          Books,        N,
,Cash^Accrual^Consolidation^Elimination,b.book in (#Book#), ,N,
//END FILTER
```

Commit your changes to the script if you expect to reuse the filter section in another YSR report design. Otherwise, continue the report-building procedures described here without committing any changes to your script. Note that the Yardi Excel Add-In does not save any elements of the built filter section when you close the **Build Your Script Filter** dialog box.



Committing your changes to your script is optional. Click **Commit Changes to Script** only if you plan to reuse the script in another YSR report and you want to save the filter section. The Yardi Excel Add-In creates a complete YSR report with all setup elements (including filter fields) whether or not you commit the filter elements to your script.

If you make changes to the filter section of your script in an external text editor such as Notepad, you can rebuild the filter elements in the Yardi Excel Add-In dialog by clicking **Rebuild Grid from Script File**.

### //Filter Section Columns

For each filter item, the Yardi Excel Add-In helps you complete the following data columns of the //Filter section of a Yardi script: Element Type, Data Type, Name, Caption, List, Val 1, Mandatory, and Multi-Select. Two filter section columns are *not* accessible when using the Yardi Excel Add-In: column 5, the key column, and column 8, the Value 2 clause. The key column is not used in conventional Yardi SQL script filters, and the Value 2 clause does not apply to YSR filtration. When you commit a built filter back to the underlying script file, the Yardi Excel Add-In inserts blank columns separated by a comma to stand in for the empty columns and thereby maintains the organizational accuracy of the filter section.

### Program Object iTypes (Column 6 and List OR Lookup Type Field)

Column 6 in the filter section of a Yardi script (associated with the **List OR Lookup Type** field on the **Build Your Script Filter** tab) supports a numeric shorthand also known as the program object iType. Not all programming objects are supported by YSR, and some lookup lists that are supported by YSR have no matching program object iType.

If you select as your data source a script with a filter section that refers to a supported object iType (for example, 3), the Yardi Excel Add-In identifies the object iType and displays the corresponding lookup list name in the **Lookup Type** field (in this example, ysiPropertyLookup (3)).

If, however, you are using an object iType that is not supported by YSR, you cannot use the **Lookup Type** option. Instead, you must author the lookup list using a syntactically complete SQL select statement.

For more information about common program object iTypes, see Chapter 5, “Vista Filters” in the *Yardi SQL Scripting Guide*. Refer to the section on Lists, since List is Column 6 of the filter section.

## Filter Mapping and the Value1 Parameter

The Value1 (Val1) parameter is Column 7 of a Yardi //Filter section. It contains a small amount of SQL code that is added (with AND) to the WHERE clause of a select statement if there is a #Conditions# operator in the //Select section. For example, the Val1 parameter for a property filter might be **p.hmy = #prop#**. If the WHERE clause of your script contains #Conditions#, then **p.hmy = #prop#** is added to the WHERE clause of the select statement at run time.



The Val1 parameter is also the clause that you use to map filters if you are setting up a YSR report manually. For more information about filter mapping, see “Filter Mapping for Scripted Report Sections” on page 81.

When you use the Yardi Excel Add-In to build the filters for your report, it supplies a Val1 parameter based on the tokens it identifies in the script. For example, if the WHERE clause of your script contains the #PROPERTY# token as a condition, the Yardi Excel Add-In supplies **p.hmy in (#Property#)** as the Val1 clause for the //Filter section.



The Yardi Excel Add-In relies on conventional table names and aliases for constructing Val1 parameters. For example, if your script contains the #PROPERTY# token, the Yardi Excel Add-In expects **Property** as the name for the underlying data table and **p** as the table alias. If your aliases do not match the conventional aliases, you can edit them either in your script or in the Yardi Excel Add-In. For more information, see “Val1 Parameter Conventions” on page 272.

## Val1 Parameter Conventions

If your script file contains one of a set of named tokens in the WHERE clause, the Yardi Excel Add-In constructs a corresponding filter field and maps the filter to the //Select section by supplying a Val1 clause (for example, **p.hmy in (#Property#)**). At run time, Voyager replaces the token in the //Select section with the Val1 parameter.

The following table provides a list of the named tokens for which the Yardi Excel Add-In constructs a Val1 clause, the Val1 clause, and the expected table naming conventions.

Named Token in the //Select section (case insensitive)	Val1 Clause	Underlying Schema Table	Table Alias
#PROPERTY#	p.hmy in (#Property#)	Property	p
#TENANT#	t.hmyPerson in (#Tenant#)	Tenant	t
#UNIT#	u.hmy in (#Unit#)	Unit	u

#OWNER#	o.hmyperson in (#Owner#)	Owner	o
#BANK#	b.hmy in (#Bank#)	Bank	b
#ACCT#	a.hmy in (#Acct#)	Acct	a
#ACCOUNT#	a.hmy in (#Account#)	Acct	a
#JOB#	j.hmy in (#Job#)	Job	j
#CATEGORY#	jc.hmy in (#Category#)	Category	jc
#UNITTYPE#	ut.hMy in (#UnitType#)	UnitType	ut
#BUILDING#	b.hMy in (#Building#)	Building	b
#COUNTRY#	ci.hmy in (#Country#)	Country	ci
#CUSTOMER#	c.hmyperson in (#Customer#)	Customer	c
#CONTACT#	c.hmy in (#Contact#)	Contact	c
#FLOOR#	f.hmy in (#Floor#)	Floors	f
#FLOORS#	f.hmy in (#Floors#)	Floors	f
#BOOK#	b.book in (#Book#)	Book	b

## Building Report Packages (Dump SQL)

The report-building (**Dump SQL**) function of the Yardi Excel Add-In creates a .pkg file that you can load into Voyager. The .pkg file contains all elements of the YSR report design except for the report template and custom SQL script. You must save the template and script to your Reports path to accompany the YSR report.

The report-building process in the Add-In is intended for quick prototyping of reports, and it makes a number of simplifications. After you build the report package and load it into Voyager, you can extend your report design using the Voyager user interface.



You can use the Yardi Excel Add-In to build reports with one sub-report only, using one SQL script file for data retrieval. The SQL script can have multiple sections, however. Each section is a uniquely named select statement, with the exception of the (optional) top-level select statement, which must be unnamed.



Currently you cannot create Dump SQL packages when using a Voyager analytics data source, but this is planned for future releases.



The Yardi Excel Add-In can, as part of its Dump SQL operations, prepare a YSR for use with Report Scheduler. For more information, see “Dump SQL Tab Screen Reference” on page 275.

If you have a .txt script file for data retrieval and an Excel template that uses smart markers to represent the data, you are ready to build a YSR report.

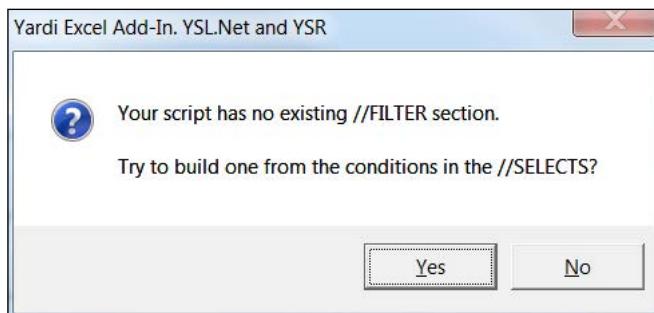
## To build a YSR report with the Yardi Excel Add-In

- 1 Open your Excel report template.
- 2 On the Excel ribbon, click the **Add-Ins** tab.
- 3 Click **Yardi Spreadsheet Reporting Setup**. The **Yardi Spreadsheet Reporting Setup** screen appears.
- 4 Select your script file as the data source for your YSR report. For more information, see “To select a custom SQL script as a data source” on page 251.



Check the **Analytic Standard Data Sources** tab to make sure that no Analytics data sources are selected. If any Analytics data sources are selected, the report-building features of the Add-In are disabled.

- 5 Click **Ok**.
- 6 On the **Add-Ins** tab, click **Dump SQL (YSR)**. The **Yardi Spreadsheet Reporting Dump SQL** screen appears. If your script file does not contain filters, the following message appears:



- a Select **Yes** or **No**.
  - 7 Complete the fields on the **Dump SQL** tab. For more information, see “Dump SQL Tab Screen Reference” on page 275.
  - 8 Build or edit your filter fields.
    - a Click the **Build Your Script Filter** tab.
    - b Complete the fields. For more information, see “Build Your Script Filter Tab Screen Reference” on page 277.
    - c If you want to reuse the script later as a data source for another YSR report, you can click **Commit Changes to Script**. The Add-In adds a filter section and a notes section to your script file.
- 
- If you do not click **Commit Changes to Script**, the Yardi Excel Add-In can still build a complete YSR report with all design elements, including filter fields.
- 9 Click the **Dump SQL** tab.
- Yardi Voyager Yardi Spreadsheet Reporting User's Guide

- 10** Review your selections and click **OK**. The Add-In exports a .pkg file containing all report setup elements to the location you specified on the **Script Data Sources** tab of the **Yardi Spreadsheet Reporting Setup** screen.



The last step is to load your package through Voyager Workstation Administration and save your Excel template and SQL script file to the Reports path. Then you are ready to generate the report.

## Dump SQL Tab Screen Reference

Dump SQL | Build your Script Filter | Build your Analytics Filter |

---

Voyager version targeted

YSR in Voyager 6008 or 7S pre Reporting Tools PI 4 (8 char)  
 YSR in Voyager 7S post Reporting Tools PI 4 (16 char sCode)

Status

---

Merge Report Code

YSR Excel Template :Copy of YSR\_OccupiedAreaOverTime.xlsx

Description

---

Top Level Select

My script has an unnamed //Select: use that.  
 Use the SQL Select Statement immediately to the right.  
 I'm not using a Top Level Select or a Key Column.

Key Column

---

Clicking OK will create a Yardi PKG file and save it in the same location as the script data file folder defined on the Yardi Spreadsheet Reporting Setup Screen. Executing it, via VW Admin, will introduce a new one, or update any existing YSR report in your Voyager database with the same name as the Merge Report Code you define above.

### Voyager Version Targeted

The Yardi Excel Add-In prepares a .pkg text file for the version of Voyager that you select.

### Merge Report Code

A unique code for your report (required).

**TIP** The Yardi Excel Add-In suggests a random code and description, but you can edit these. Do not use common codes like TrialBal or Budget because the execution of the package created by the Yardi Excel Add-In will, without warning, overwrite any existing YSR report with the same code. To make sure that you will not overwrite an existing YSR report, log in to Voyager and check to make sure that there is no other YSR report (active or inactive) with the same report code.

---

<b>Top-Level Select section</b>	Options for handling the top-level select statement.
	<p><b>My script has an unnamed //Select: use that</b> Indicates that your script file has an unnamed select statement. This select becomes the top-level select. Enter the primary key of the top-level select in the <b>Key Column</b> field to the right.</p> <p><b>Use the SQL Select Statement immediately to the right</b> Enter a select statement in the field to the right. You must also specify the primary key in the <b>Key Column</b> field.</p> <p><b>I'm not using a top-level Select or a Key Column</b> Select this option if you are not using a top-level select statement.</p> <p>For more information about the top-level select statement, see “Report Design Considerations” on page 23.</p>
<b>Key Column</b>	<p>The name of the column that uniquely identifies the data that Voyager displays on the report generation screen, and the relational column for linking sub-report data.</p> <p><b>NOTE</b> You must complete this field if you are using a top-level select statement.</p> <p>Voyager uses this column to link data from multiple tables. For example, a table containing property data typically has a column that provides unique property codes for each property. If you want to use the property code to relate information from multiple tables, enter the alias of the property code column (the unique identifier) here.</p> <p>For more information, see “Identifying Your Key Column” on page 25.</p> <p><b>NOTE</b> As a best practice, do not use ‘property’ as the alias for the key column. Key columns aliased ‘property’ can cause reports to display no data for users with property list security. This applies to all aliases for all key columns in the report.</p>
<b>Create matching script file for Report Scheduler</b>	<p>This option appears if you are using a top-level select statement.</p> <p>When selected, the Yardi Excel Add-In creates a PKG file that accords with the content structure and naming conventions of Report Scheduler. For more information about Report Scheduler, see “Report Scheduler” on page 151.</p>

---

## Build Your Script Filter Tab Screen Reference

**Commit Changes to Script** Click to edit your .txt script file. The Yardi Excel Add-In writes your filter elements to your script, adds a //Notes section, and saves the changes.

**Rebuild Grid from Script File** Click to import filter elements from your script to the Yardi Excel Add-In.

**Element Type** The type of filter element.

For more information about filter elements, see Chapter 5, “Vista Filters” in the *Yardi SQL Scripting Guide*.

**Data Type** The type of data the filter accepts.

For more information about data types, see Chapter 5, “Vista Filters” in the *Yardi SQL Scripting Guide*.

**Name** The name of the filter field. This value is analogous to the **Field Name** on the **Report Filters Setup** screen.

**Caption** The label of the filter field, visible to users on the report generation screen.

**List OR Lookup Type** Use the **List** field to create a custom list, or use the **Lookup Type** field to select an existing lookup list, but do not use both fields.

In the **List** field, you can create a custom list with values separated by carets (^) or you can enter a syntactically complete SQL select statement. YSR processes the select statement and displays the results to the user in a drop-down list.

If you chose an existing lookup list from the **Lookup Type** field, YSR executes the underlying YSI.Net lookup and presents the results to the user as a pick-list.

---

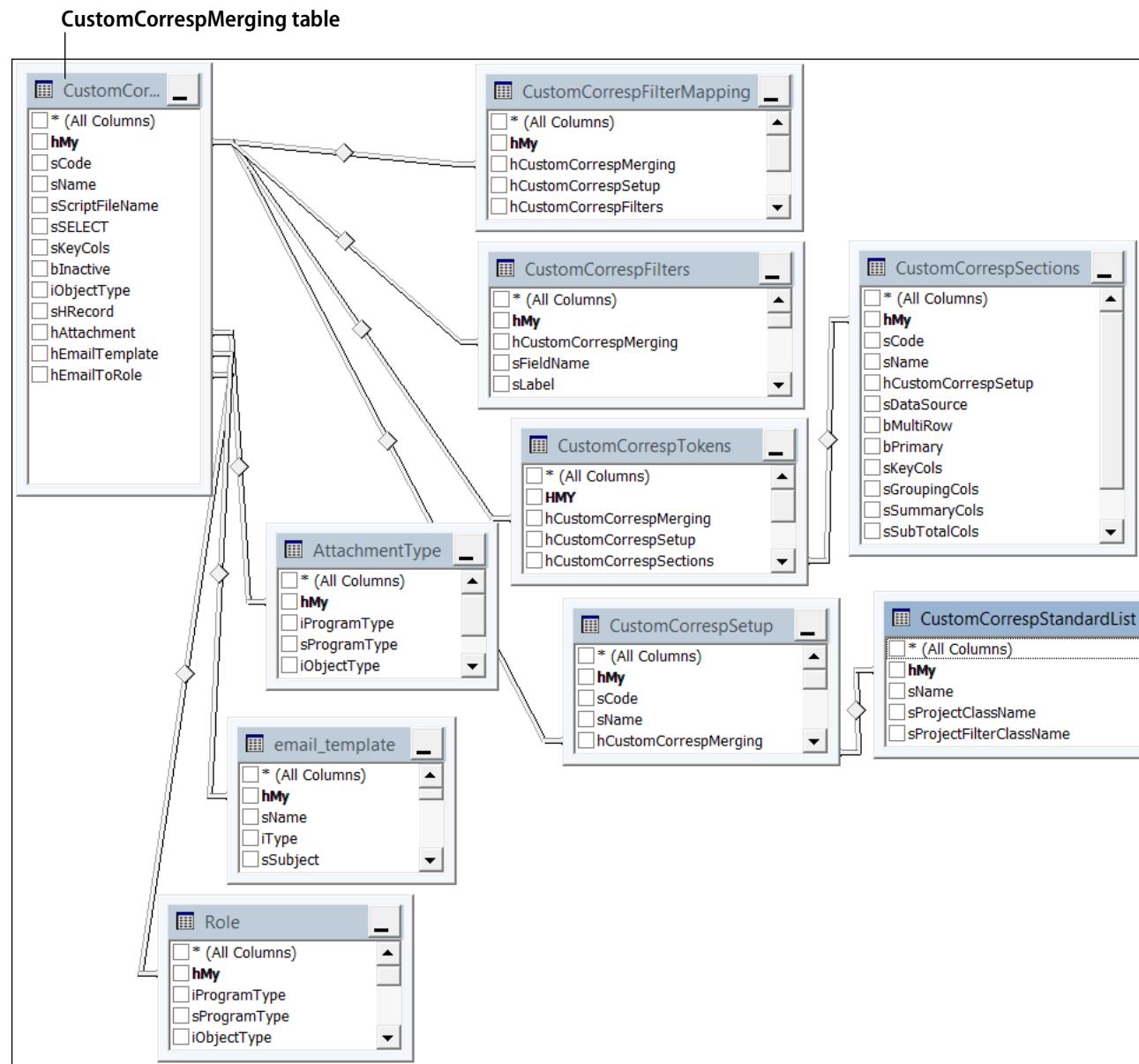
<b>Val1</b>	A small section of SQL appended to the executing script data source to introduce the filtration specified by the user.  For example, if you are building a filter element in this row named Property, and you have a named token in your script data source in a WHERE clause (WHERE 1=1 #Property#), and you have aliased the Property table in your SELECT statement as p, then a suitable Value1 clause is: <b>p.hmy in (#Property#)</b> .  <b>TIP</b> Value1 clauses are appended to the WHERE clause with the word AND. Therefore do not use the word AND as the first word in a Value1 clause.  <b>NOTE</b> If your data source contains one of the set of tokens known to the Yardi Excel Add-In, the Yardi Excel Add-In attempts to build the Value1 clause for you. For more information about how the Yardi Excel Add-In constructs Value1 clauses, see "Filter Mapping and the Value1 Parameter" on page 272.  For more information about Value1 clauses, see Chapter 5, "Vista Filters" in the <i>Yardi SQL Scripting Guide</i> .
<b>Mandatory</b>	Makes the filter field a required field.
<b>Multi-Select</b>	Users can filter for multiple items.
<b>Parent</b>	Indicates that this filter field is restricted by values in another filter field. Enter the sequence number of the parent filter field.
<b>Automatically Rename Named Tokens to Numbered Conditions in Script File on Dump SQL (recommended on first use)</b>	Replaces named tokens in your data source with tokens in the #Conditionn# format (#Condition1#, #Condition2#, and so on).  For example, if you select this check box, the Yardi Excel Add-In makes this replacement when you click <b>Dump SQL</b> :  WHERE 1=1 #Property# #Tenant# becomes  WHERE 1=1 #Condition1# #Condition2#  <b>TIP</b> All versions of YSR require the #Conditionn# format in the WHERE clause, rather than the named token format.

---

## APPENDIX A

# YSR Report Schema

When you create a YSR report, Voyager inserts a unique record into the CustomCorrespMerging table. The following image is a graphical representation of the main data structures, or schema, of a YSR report. Additional tables (not shown here) support optional features like digital signatures.



## APPENDIX B

# YSR Reports and Component Files by Vertical

This section lists the standard YSR reports and their component files (top-level SQL script files, sub-report script files, and template files) by vertical.

## 1099

### 1096 Vendor - by Bank (1096VBB)

*Top-Level Select Script Files*

rs\_1096.txt

*YSR Script Files*

rs\_1096.txt

COPYA

*YSR Template Names*

ydoc\_1096\_IRSCopy.docx

COPYA

### 1096 Vendor - by Mgmt. Co. (1096VBM)

*Top-Level Select Script Files*

rs\_1096.txt

*YSR Script Files*

rs\_1096.txt

COPYA

*YSR Template Names*

ydoc\_1096\_IRSCopy.docx

COPYA

### 1096 Vendor - by Owner (1096VBO)

*Top-Level Select Script Files*

rs\_1096.txt

*YSR Script Files*

rs\_1096.txt

COPYA

*YSR Template Names*

ydoc\_1096\_IRSCopy.docx

COPYA

### 1096-MISC Owner - by Mgmt. Co. (1096OBM)

*Top-Level Select Script Files*

rs\_1096.txt

*YSR Script Files*

rs\_1096.txt

COPYA

*YSR Template Names*

ydoc\_1096\_IRSCopy.docx

COPYA

**1098 Report YSR (1098\_YSR)**

*Top-Level Select Script Files*

rs\_sql\_1098-Coop.YSR\_Form.txt

*YSR Script Files*

rs\_sql\_1098-Coop.YSR\_Form.txt

1098\_YSR

*YSR Template Names*

select 'YSR\_1098\_' + (Case '#ReportType#' when 'Report Only' then 'Report\_Template.xlsx' when 1098\_YSR

'Print Forms-Copy A' then 'Form\_CopyA\_Template.docx' when 'Print Forms-Copy B' then 'Form\_CopyB\_Template.docx' when 'Print Forms-Copy C' then 'Form\_CopyC\_Template.docx' when 'Form 1096' then 'Form\_1096\_IRSCopy.docx' end)

**1099-DIV Dividends by Management (1099DBM)**

*Top-Level Select Script Files*

rs\_1099\_MISCnINT.txt

*YSR Script Files*

rs\_1099\_MISCnINT.txt

COPYB

rs\_1099\_MISCnINT.txt

COPYB2

rs\_1099\_MISCnINT.txt

COPYC

rs\_1099\_MISCnINT.txt

COPYC2

*YSR Template Names*

ydoc\_1099DIV\_CopyB.docx

COPYB

ydoc\_1099DIV\_CopyB2.docx

COPYB2

ydoc\_1099DIV\_CopyC.docx

COPYC

ydoc\_1099DIV\_CopyC2.docx

COPYC2

**1099-DIV Dividends by Owner (1099DBO)**

*Top-Level Select Script Files*

rs\_1099\_MISCnINT.txt

*YSR Script Files*

rs\_1099\_MISCnINT.txt

COPYB

rs\_1099\_MISCnINT.txt

COPYB2

rs\_1099\_MISCnINT.txt

COPYC

rs\_1099\_MISCnINT.txt

COPYC2

*YSR Template Names*

ydoc_1099DIV_CopyB.docx	COPYB
ydoc_1099DIV_CopyB2.docx	COPYB2
ydoc_1099DIV_CopyC.docx	COPYC
ydoc_1099DIV_CopyC2.docx	COPYC2

**1099-INT Interest - by Mgmt. Co. (1099ITIM)***Top-Level Select Script Files*

rs\_1099\_MISCnINT.txt

*YSR Script Files*

rs_1099_MISCnINT.txt	COPYA
rs_1099_MISCnINT.txt	COPYB
rs_1099_MISCnINT.txt	COPYB2
rs_1099_MISCnINT.txt	COPYC
rs_1099_MISCnINT.txt	COPYC2

*YSR Template Names*

ydoc_1099INT_CopyA.docx	COPYA
ydoc_1099INT_CopyB.docx	COPYB
ydoc_1099INT_CopyB2.docx	COPYB2
ydoc_1099INT_CopyC.docx	COPYC
ydoc_1099INT_CopyC2.docx	COPYC2

**1099-INT Interest - by Owner (1099ITIO)***Top-Level Select Script Files*

rs\_1099\_MISCnINT.txt

*YSR Script Files*

rs_1099_MISCnINT.txt	COPYA
rs_1099_MISCnINT.txt	COPYB
rs_1099_MISCnINT.txt	COPYB2
rs_1099_MISCnINT.txt	COPYC
rs_1099_MISCnINT.txt	COPYC2

*YSR Template Names*

ydoc_1099INT_CopyA.docx	COPYA
ydoc_1099INT_CopyB.docx	COPYB
ydoc_1099INT_CopyB2.docx	COPYB2
ydoc_1099INT_CopyC.docx	COPYC
ydoc_1099INT_CopyC2.docx	COPYC2

**1099-MISC Owner - by Mgmt. Co. (1099MOBM)***Top-Level Select Script Files*

rs\_1099\_MISCnINT.txt

*YSR Script Files*

rs_1099_MISCnINT.txt	COPYA
rs_1099_MISCnINT.txt	COPYB
rs_1099_MISCnINT.txt	COPYB2
rs_1099_MISCnINT.txt	COPYC
rs_1099_MISCnINT.txt	COPYC2

*YSR Template Names*

ydoc_1099MISC_CopyA.docx	COPYA
ydoc_1099MISC_CopyB.docx	COPYB
ydoc_1099MISC_CopyB2.docx	COPYB2
ydoc_1099MISC_CopyC.docx	COPYC
ydoc_1099MISC_CopyC2.docx	COPYC2

**1099-MISC Vendor - by Bank A/c Vendor (1099MVBB)***Top-Level Select Script Files*

rs\_1099\_MISCnINT.txt

*YSR Script Files*

rs_1099_MISCnINT.txt	COPYA
rs_1099_MISCnINT.txt	COPYB
rs_1099_MISCnINT.txt	COPYB2
rs_1099_MISCnINT.txt	COPYC
rs_1099_MISCnINT.txt	COPYC2

*YSR Template Names*

ydoc_1099MISC_CopyA.docx	COPYA
ydoc_1099MISC_CopyB.docx	COPYB
ydoc_1099MISC_CopyB2.docx	COPYB2
ydoc_1099MISC_CopyC.docx	COPYC
ydoc_1099MISC_CopyC2.docx	COPYC2

**1099-MISC Vendor - by Mgmt. Co. (1099MVBM)***Top-Level Select Script Files*

rs\_1099\_MISCnINT.txt

*YSR Script Files*

rs_1099_MISCnINT.txt	COPYA
rs_1099_MISCnINT.txt	COPYB
rs_1099_MISCnINT.txt	COPYB2
rs_1099_MISCnINT.txt	COPYC
rs_1099_MISCnINT.txt	COPYC2

*YSR Template Names*

ydoc_1099MISC_CopyA.docx	COPYA
ydoc_1099MISC_CopyB.docx	COPYB
ydoc_1099MISC_CopyB2.docx	COPYB2

ydoc_1099MISC_CopyC.docx	COPYC
ydoc_1099MISC_CopyC2.docx	COPYC2

**1099-MISC Vendor - by Owner (1099MVBO)***Top-Level Select Script Files*

rs\_1099\_MISCnINT.txt

*YSR Script Files*

rs_1099_MISCnINT.txt	COPYA
rs_1099_MISCnINT.txt	COPYB
rs_1099_MISCnINT.txt	COPYB2
rs_1099_MISCnINT.txt	COPYC
rs_1099_MISCnINT.txt	COPYC2

*YSR Template Names*

ydoc_1099MISC_CopyA.docx	COPYA
ydoc_1099MISC_CopyB.docx	COPYB
ydoc_1099MISC_CopyB2.docx	COPYB2
ydoc_1099MISC_CopyC.docx	COPYC
ydoc_1099MISC_CopyC2.docx	COPYC2

**Affordable****Affordable 50059 Review (Aff50059Review)***Top-Level Select Script Files**YSR Script Files*

rs\_AFF\_Review59\_YSR.txt Review

*YSR Template Names*

rx\_AFF\_Review59\_YSR.xlsx Review

**Affordable Daily Activity Report (AffDailyActivity)***Top-Level Select Script Files**YSR Script Files*

rs\_aff\_YSR\_Daily\_Activity.txt DailyAct

*YSR Template Names*

rs\_aff\_YSR\_Daily\_Activity.xlsx DailyAct

**Affordable Database Health Diagnostic Report (AFFDBHDR)***Top-Level Select Script Files**YSR Script Files*

rs\_Aff\_DB\_Health\_Diagnostic\_Report\_2.txt

AFFDBHDR

<i>YSR Template Names</i>	
rx_Aff_DB_Health_Diagnostic_Report_2.xlsx	
AFFDBHDR	
<b>Affordable GPR Report (AffGPR)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_Aff_YSR_Report_OutputType.TXT	affgpr
<i>YSR Template Names</i>	
Select 'ysr_Aff_GPR_By' + '#SummarizeBy#' + '.xlsx'	affgpr
<b>Affordable NY Tax Credit and HOME Rent Roll Report (AFFPSRNY)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_Aff_TaxCr_PSR_NY_HPD.txt	
AFFPSRNY	
<i>YSR Template Names</i>	
rx_Aff_TaxCr_PSR_NY_HPD.xls	
AFFPSRNY	
<b>Affordable Rent Roll (AffRentRoll)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_Aff_YSR_Report_OutputType.TXT	affrroll
<i>YSR Template Names</i>	
Select 'ysr_Aff_RentRoll_' + REPLACE('#SummarizeBy#', ' ', '') + '.xlsx'	affrroll
<b>Affordable Rent Roll with Lease Charges (AffRntRollLsChgs)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_Aff_YSR_Report_OutputType.TXT	rrlschgs
<i>YSR Template Names</i>	
Select 'ysr_Aff_RentRollLsChgs_' + REPLACE('#SummarizeBy#', ' ', '') + '.xlsx'	rrlschgs
<b>Affordable Special Claim Candidates Report (AffSplClaimCand)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_Aff_SpecialClaim_Candidate.txt	AffClaim
<i>YSR Template Names</i>	
rx_Aff_SpecialClaim_Candidate.xlsx	AffClaim
<b>Affordable Tenant Data Collection (AFFTDC)</b>	

*Top-Level Select Script Files**YSR Script Files*

rs\_Aff\_TIC\_TenantDataCollectionCityOfChicago.txt

AFFTDC

*YSR Template Names*

rx\_Aff\_TIC\_TenantDataCollectionCityOfChicago.xls

AFFTDC

**Affordable Tenant Demographics (AffDemo)***Top-Level Select Script Files**YSR Script Files*

rs\_Aff\_YSR\_Demographics.txt

AffDemo

*YSR Template Names*

rx\_Aff\_YSR\_Demographics.xlsx

AffDemo

**Affordable Unit Vacancy (AffUnitVacancy)***Top-Level Select Script Files**YSR Script Files*

rs\_Aff\_YSR\_Report\_OutputType.TXT

affuvac

*YSR Template Names*

Select 'ysr\_Aff\_UnitVacancy\_' + '#SummarizeBy#' + '.xlsx'

affuvac

**Annual Income Qualified Unit Report (AFFCPSRNY)***Top-Level Select Script Files**YSR Script Files*

rs\_Aff\_TaxCr\_PSR\_NY.txt

AFFPSRNY

*YSR Template Names*

rx\_Aff\_TaxCr\_PSR\_NY.xls

AFFPSRNY

**Audit NAHMA Fields (AffNAHMAAudit)***Top-Level Select Script Files**YSR Script Files*

rs\_Aff\_AuditNAHMAFields.txt

Assets

rs\_Aff\_AuditNAHMAFields.txt

EventDet

rs\_Aff\_AuditNAHMAFields.txt

HHMem

rs\_Aff\_AuditNAHMAFields.txt

Income

*YSR Template Names*

rx\_Aff\_AuditNAHMAFields.xlsx

Assets

rx\_Aff\_AuditNAHMAFields.xlsx

EventDet

rx\_Aff\_AuditNAHMAFields.xlsx

HHMem

rx_Aff_AuditNAHMAFields.xlsx	Income
<b>Database Snapshot Report (DBSnapRp)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_Aff_YSR_DatabaseSnapshot.txt	
<b>DBSnapRp</b>	
<i>YSR Template Names</i>	
rx_Aff_YSR_DatabaseSnapshot_template.xlsx	
<b>DBSnapRp</b>	
<b>DC PSR Building Status (AFFTCPSRDC)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_Aff_TaxCr_PSR_DC.txt	TCPSRDC
<i>YSR Template Names</i>	
rx_Aff_TaxCr_PSR_DC.xlsx	TCPSRDC
<b>HOME Rent and Occupancy Report (AffHMRO)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_Aff_HOME_RentOcc.txt	
<b>HMRNTOCC</b>	
<i>YSR Template Names</i>	
rx_Aff_HOME_RentOcc.xls	
<b>HMRNTOCC</b>	
<b>Local Program Unit Detail (LPUnitMapping)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_Aff_LocalProgram_UnitMapping.txt	UnitMap
<i>YSR Template Names</i>	
rx_Aff_LocalProgram_UnitMapping.xlsx	UnitMap
<b>Minnesota CHART (AFFTCPSRMNCHART)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_Aff_TaxCr_PSR_MN_CHART.txt	
<b>MNCHART</b>	
<i>YSR Template Names</i>	
rx_aff_taxcr_psr_MN_CHART.xlsm	

**MNCHART****NYC HDC Rent Roll-Deep Rent Skew Project***Top-Level Select Script Files**YSR Script Files*

rs\_Aff\_YSR\_TaxCr\_NYC\_HDC\_RentRoll\_DeepRentSkew\_Project.txt

HPDRSPR

*YSR Template Names*

rx\_Aff\_YSR\_TaxCr\_NYC\_HDC\_RentRoll\_DeepRentSkew\_Project.xlsx

HPDRSPR

**NYC HDC Rent Roll-Hundred Percent LIHTC Project***Top-Level Select Script Files**YSR Script Files*

rs\_Aff\_YSR\_TaxCr\_NYC\_HDC\_RentRoll\_HundredPercentLIHTC\_Project.txt

HPLIHTC

*YSR Template Names*

rx\_Aff\_YSR\_TaxCr\_NYC\_HDC\_RentRoll\_HundredPercentLIHTC\_Project.xlsx

HPLIHTC

**NYC HDC Rent Roll-Mixed Income Project (AFFTCNYCHDCMIXIN)***Top-Level Select Script Files**YSR Script Files*

rs\_Aff\_YSR\_TaxCr\_NYC\_HDC\_RentRoll\_MixedIncome\_Project.txt

HPMIXINC

*YSR Template Names*

rx\_Aff\_YSR\_TaxCr\_NYC\_HDC\_RentRoll\_MixedIncome\_Project.xlsx

HPMIXINC

**NYC HDC Rent Roll-NYCHA Mixed Income Project***Top-Level Select Script Files**YSR Script Files*

rs\_Aff\_YSR\_TaxCr\_NYC\_HDC\_RentRoll\_NYCHA\_MixedIncome\_Project.txt

HPMIXINC

*YSR Template Names*

rx\_Aff\_YSR\_TaxCr\_NYC\_HDC\_RentRoll\_NYCHA\_MixedIncome\_Project.xlsx

HPMIXINC

**NYC HPD Special Rent Roll (AFFTCNYCHPDSPRLRR)***Top-Level Select Script Files**YSR Script Files*

rs\_Aff\_YSR\_TaxCr\_NYC\_HPD\_RentRoll.txt

HPDSPLRR

*YSR Template Names*

rx\_Aff\_YSR\_TaxCr\_NYC\_HPD\_RentRoll.xlsx

HPDSPLRR

## Canadian Social Housing

### Additional Income Verification Letter 2, RGI Nova Scotia

*Top-Level Select Script Files*

rs\_RGI\_NS\_Notice\_VerifyInc2\_ysr.txt

*YSR Script Files*

rs\_RGI\_NS\_Notice\_VerifyInc2\_ysr.txt

Addition

*YSR Template Names*

Rx\_RGI\_Notice\_VerifyInc2.docx

Addition

### Application for Permission to keep a Pet (PetApp)

*Top-Level Select Script Files*

*YSR Script Files*

rs\_NS\_PetApplication.txt

PetApp

*YSR Template Names*

ydoc\_NS\_PetApplication.docx

PetApp

### Fax document, Fire Alarm Connect Phone Number, RGI NS

*Top-Level Select Script Files*

*YSR Script Files*

rs\_NS\_ConnectFireAlarm.txt

FirAlarm

*YSR Template Names*

ydoc\_NS\_ConnectFireAlarm.docx

FirAlarm

### Fax document, Fire Alarm Disconnect Phone Number, RGI NS

*Top-Level Select Script Files*

*YSR Script Files*

rs\_NS\_DisconnectFireAlarm.txt

DiscFire

*YSR Template Names*

ydoc\_NS\_DisconnectFireAlarm.docx

DiscFire

### Lease Amendment Letter, RGI Nova Scotia (NSLeaseAmendmt)

*Top-Level Select Script Files*

*YSR Script Files*

rs\_RGI\_NS\_LesseeAmendment\_ysr.txt

NSLA1

*YSR Template Names*

y\_RGI\_NS\_LesseeAmendment.docx

NSLA1

**Notice to Quit, RGI Nova Scotia (QuitNotice)***Top-Level Select Script Files**YSR Script Files*

rs\_RGI\_NS\_QuitNotice.txt

QuitNot

*YSR Template Names*

y\_RGI\_NS\_QuitNotice.docx

QuitNot

**NS Standard Lease Form (NSStdLease)***Top-Level Select Script Files*

rs\_RGI\_NS\_LeaseStdForm\_YSR.txt

*YSR Script Files*

rs\_RGI\_NS\_LeaseStdForm\_YSR.txt

NSStdLea

*YSR Template Names*

LeaseDocument\_Template.docx

NSStdLea

**Property Owner Letter, RGI Nova Scotia (PropertyOwnerLet)***Top-Level Select Script Files**YSR Script Files*

rs\_RGI\_NS\_PropertyOwnerLetter\_ysr.txt

PropOwnL

*YSR Template Names*

y\_RGI\_NS\_PropertyOwnerLetter.docx

PropOwnL

**Transfer Form, RGI Nova Scotia (TransferForm)***Top-Level Select Script Files**YSR Script Files*

rs\_YSR\_NS\_TransferForm.txt

Transfrm

*YSR Template Names*

ydoc\_NS\_TransferForm\_Crystal.docx

Transfrm

**Commercial****Commercial Rent Roll (RentRoll)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_Property\_rentroll.xlsx

RentRoll

**Contact Directory (Contact)**

*Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_Comm\_ContactDirectory.xlsx

Contact

**Critical Dates (CriDates)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_Comm\_CriticalDates.xlsx

CriDates

**CRM Analytics - Consolidated Report (CRMAnaly)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

SELECT 'yDoc\_CRMAnalytics' + (CASE '#ShowUnassignedLeads#' WHEN 'Yes' THEN CASE

CRMAnaly

```
'#SummaryType#' WHEN 'Property' THEN '_LA_Lead_Property' WHEN 'Leasing Agent' THEN
'_LA_Lead_LA' WHEN 'Customer' THEN '_LA_Lead_Cust' END WHEN 'No' THEN CASE
'#SummaryType#' WHEN 'Property' THEN '_Prop' WHEN 'Leasing Agent' THEN '_LA' WHEN
'Customer' THEN '_Cust' END END)+'.xlsx'
```

**Customer Detail (CustDet)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_Comm\_CustomerDetail.xlsx

CustDet

**Customer Directory (CustDir)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_Lease\_CustomerDirectory.xlsx

CustDir

**Customer Top X (CustTopX)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_Comm\_CustomerTopX.xlsx

TopX

**Deal Directory (DealDir)**

*Top-Level Select Script Files**YSR Script Files*

YSR\_Lease\_DealDirectory.txt

DealDir

*YSR Template Names*

Select 'yDoc\_Lease\_DealDirectory\_' + '#SummarizeBy#' + '.xlsx'

DealDir

**Lease Activity History (LeActHis)***Top-Level Select Script Files**YSR Script Files*

YSR\_KPI\_LeasingActivityHistory.txt

ActHis

*YSR Template Names*

Select 'yDoc\_KPI\_LeasingActivityHistory\_' + '#Period#' + '#SummarizeBy#' + '.xlsx'

ActHis

**Option Expiration (OptExp)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_Comm\_OptionExpiration.xlsx

OptExp

**Portfolio Summary (PortSmry)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_KPI\_PortfolioSummary.xlsx

PortSmry

**Property Summary (PropSmry)***Top-Level Select Script Files**YSR Script Files*

YSR\_KPI\_PropertySummary.txt

PropSmry

*YSR Template Names*

Select 'yDoc\_KPI\_PropertySummary\_' + '#SummarizeBy#' + '.xlsx'

PropSmry

**Straight-Line By Lease Summary (STLSumm)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_Comm\_StraightLineByLeaseSummary.xlsx

STLSumm

**Straight-Line JE Detail (STLJEDet)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_Comm\_StraightlineRentsJEDetail.xlsx

STLJEDet

**Straight-Line JE Register (STLJEReg)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_Comm\_StraightlineRentsJERegister.xlsx

STLJEReg

**Tenancy Schedule (Tenancy)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_TenancySchedule.xlsx

Tenancy

**Unpaid Charges (UnpChg)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

yDoc\_Comm\_UnpaidCharges.xlsx

UnpChg

## Concierge

**Concierge Parking Pass (CCGParkingPass)***Top-Level Select Script Files*

rs\_CCGParkingPass\_YSR.txt

*YSR Script Files*

rs\_CCGParkingPass\_YSR.txt

ParkingP

*YSR Template Names*

CCGParkingPass\_YSR.xlsx

ParkingP

**Concierge Parking Pass Custom (CCGParkingCustom)***Top-Level Select Script Files**YSR Script Files*

rs\_CCGParkingPassCustom.txt

ParkingP

<i>YSR Template Names</i>	
rs_CCGParkingPassCustom.xlsx	ParkingP
<b>Concierge Passes (CCGPass)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
RS_CCGPass_YSR.txt	CCGPass
<i>YSR Template Names</i>	
Select Case '#PrintPopup#' when 'Yes' then 'CCGPass_YSR.xlsx' when 'No' then 'CCGPassNoPopup_YSR.xlsx' End	CCGPass
<b>Concierge SMS sent to the residents. (ConciergeSMS)</b>	
<i>Top-Level Select Script Files</i>	
YSR_ConciergeSMS.txt	
<i>YSR Script Files</i>	
YSR_ConciergeSMS.txt	ccgsmss
<i>YSR Template Names</i>	
YSR_ConciergeSMS.docx	ccgsmss
<b>Concierge Visitor (CCGVisitor)</b>	
<i>Top-Level Select Script Files</i>	
rs_CCGVisitor_YSR.txt	
<i>YSR Script Files</i>	
rs_CCGVisitor_YSR.txt	Visitor
<i>YSR Template Names</i>	
CCGVisitor_YSR.xlsx	Visitor

## Condo

### Assoc Arrears Notice Removal From PAP (AssocNoticeRemov)

<i>Top-Level Select Script Files</i>	
rs_AssocLetter_YSR.txt	
<i>YSR Script Files</i>	
rs_AssocLetter_YSR.txt	letter
<i>YSR Template Names</i>	
yDoc_AssocArrearsNoticeRemovalFromPAP.doc	letter

### Assoc Arrears Notice Removal From PAP Scripted

<i>Top-Level Select Script Files</i>	
rs_AssocArrearsNoticeRemovalFromPAPSCR_YSR.txt	

*YSR Script Files*

rs\_AssocArrearsNoticeRemovalFromPAPSCR\_YSR.txt letter

*YSR Template Names*

yDoc\_AssocArrearsNoticeRemovalFromPAPSCR.doc letter

**Assoc Arrears Reminder Letter (AssocArrears)***Top-Level Select Script Files*

rs\_AssocLetter\_YSR.txt

*YSR Script Files*

rs\_AssocLetter\_YSR.txt letter

*YSR Template Names*

yDoc\_AssocArrearsReminderLetter.doc letter

**Assoc Arrears Reminder Letter Scripted (AssocArrearsSCR)***Top-Level Select Script Files*

rs\_AssocLetterSCR\_YSR.txt

*YSR Script Files*

rs\_AssocLetterSCR\_YSR.txt letter

*YSR Template Names*

yDoc\_AssocArrearsReminderLetterSCR.doc letter

**Assoc Arrears Reminder NSF Letter (AssocArrearsNSF)***Top-Level Select Script Files*

rs\_AssocArrearsReminderNSFLetter\_YSR.txt

*YSR Script Files*

rs\_AssocArrearsReminderNSFLetter\_YSR.txt letter

*YSR Template Names*

yDoc\_AssocArrearsReminderNSFLetter.doc letter

**Assoc Arrears Reminder NSF Letter Scripted (AssocNSFSCR)***Top-Level Select Script Files*

rs\_AssocArrearsReminderNSFLetterSCR\_YSR.txt

*YSR Script Files*

rs\_AssocArrearsReminderNSFLetterSCR\_YSR.txt letter

*YSR Template Names*

yDoc\_AssocArrearsReminderNSFLetterSCR.doc letter

**Assoc Arrears Reminder3 Caveat Letter (AssocCaveat)***Top-Level Select Script Files*

rs\_AssocLetter\_YSR.txt

*YSR Script Files*

rs_AssocLetter_YSR.txt	letter
<i>YSR Template Names</i>	
yDoc_AssocArrearsReminderCaveatLetter.doc	letter
<b>Assoc Arrears Reminder3 Caveat Letter Scripted</b>	
<i>Top-Level Select Script Files</i>	
rs_AssocArrearsReminderCaveatLetterSCR.txt	
<i>YSR Script Files</i>	
rs_AssocArrearsReminderCaveatLetterSCR.txt	letter
<i>YSR Template Names</i>	
yDoc_AssocArrearsReminderCaveatLetterSCR.doc	letter
<b>Assoc Error On Cheque Letter (AssocErrorChk)</b>	
<i>Top-Level Select Script Files</i>	
rs_AssocErrorOnChequeLetter_YSR.txt	
<i>YSR Script Files</i>	
rs_AssocErrorOnChequeLetter_YSR.txt	letter
<i>YSR Template Names</i>	
yDoc_AssocErrorOnChequeLetter.doc	letter
<b>Assoc Error On Cheque Letter Scripted (AssocErrorChkSCR)</b>	
<i>Top-Level Select Script Files</i>	
rs_AssocErrorOnChequeLetterSCR_YSR.txt	
<i>YSR Script Files</i>	
rs_AssocErrorOnChequeLetterSCR_YSR.txt	letter
<i>YSR Template Names</i>	
yDoc_AssocErrorOnChequeLetterSCR.doc	letter
<b>Assoc Foreclosure Notice Letter (AssocForeclosure)</b>	
<i>Top-Level Select Script Files</i>	
rs_AssocLetter_YSR.txt	
<i>YSR Script Files</i>	
rs_AssocLetter_YSR.txt	letter
<i>YSR Template Names</i>	
yDoc_AssocForeclosureNoticeLetter.doc	letter
<b>Assoc Foreclosure Notice Letter Scripted (AssocForeclosSCR)</b>	
<i>Top-Level Select Script Files</i>	
rs_AssocForeclosureNoticeLetterSCR_YSR.txt	
<i>YSR Script Files</i>	
rs_AssocForeclosureNoticeLetterSCR_YSR.txt	letter

<i>YSR Template Names</i>		
yDoc_AssocForeclosureNoticeLetterSCR.doc		letter
<b>Assoc Welcome Condo Letter (AssocWelcomeC)</b>		
<i>Top-Level Select Script Files</i>		
rs_AssocWelcomeCondoLetter_YSR.txt		
<i>YSR Script Files</i>		
rs_AssocWelcomeCondoLetter_YSR.txt		letter
<i>YSR Template Names</i>		
yDoc_AssocWelcomeCondoLetter.doc		letter
<b>Assoc Welcome Condo Letter Scripted (AssocWelcomeCSR)</b>		
<i>Top-Level Select Script Files</i>		
rs_AssocWelcomeCondoLetterSCR_YSR.txt		
<i>YSR Script Files</i>		
rs_AssocWelcomeCondoLetterSCR_YSR.txt		letter
<i>YSR Template Names</i>		
yDoc_AssocWelcomeCondoLetterSCR.doc		letter
<b>Assoc Welcome HOA Letter (AssocWelcome)</b>		
<i>Top-Level Select Script Files</i>		
rs_AssocWelcomeHOALetter_YSR.txt		
<i>YSR Script Files</i>		
rs_AssocWelcomeHOALetter_YSR.txt		letter
<i>YSR Template Names</i>		
yDoc_AssocWelcomeHOALetter.doc		letter
<b>Assoc Welcome HOA Letter Scripted (AssocWelcomeSCR)</b>		
<i>Top-Level Select Script Files</i>		
rs_AssocWelcomeHOALetterSCR_YSR.txt		
<i>YSR Script Files</i>		
rs_AssocWelcomeHOALetterSCR_YSR.txt		letter
<i>YSR Template Names</i>		
yDoc_AssocWelcomeHOALetterSCR.doc		letter
<b>Condo Accounts Receivable Aging (CondoAR)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
YSR_CondoAR.xlsx		CondoAR

**Condo Analytic Owner Ledger with Aging (Analyticledger)***Top-Level Select Script Files*

rs\_AssocOwnerAnalyticLedger\_WithAging.txt

*YSR Script Files*

rs\_AssocOwnerAnalyticLedger\_WithAging.txt

ledger

*YSR Template Names*

YSR\_OwnerLedgerAnalytic\_Withaging.xlsx

ledger

**Condo Analytic Owner Ledger without Aging (AnalyticLedger1)***Top-Level Select Script Files*

rs\_AssocOwnerAnalyticLedger\_Withoutaging.txt

*YSR Script Files*

rs\_AssocOwnerAnalyticLedger\_Withoutaging.txt

ledger

*YSR Template Names*

YSR\_OwnerLedgerAnalytic\_Withoutaging.xlsx

ledger

**Condo Correspondence Letter in YSR (CondoLetter)***Top-Level Select Script Files*

rs\_AssocLetterScripted\_YSR.txt

*YSR Script Files*

rs\_AssocLetterScripted\_YSR.txt

letter

*YSR Template Names*

YSR\_AssocLetterScripted.xlsx

letter

**Condo Validation Details (cndvalidation)***Top-Level Select Script Files*

rs\_sql\_CondoValidatePropertyYSR.txt

*YSR Script Files*

rs\_sql\_CondoValidatePropertyYSR.txt

cvalid

*YSR Template Names*

YSR\_CondoValidationDetail.xlsx

cvalid

**Correspondence Condo Invoice (CondoInvoice)***Top-Level Select Script Files*

rs\_AssocInvoice.txt

*YSR Script Files*

rs\_AssocInvoice.txt

Invoice

*YSR Template Names*

YSR\_AssocInvoice.xlsx

Invoice

**Correspondence Condo Ledger (cndlledger)**

*Top-Level Select Script Files*

rs\_AssocLedger\_YSR.txt

*YSR Script Files*

rs\_AssocLedger\_YSR.txt

cledger

*YSR Template Names*

YSR\_AssocLedger.xlsx

cledger

**Correpondence Condo Letter (CndLetter)***Top-Level Select Script Files*

rs\_AssocLetter\_YSR.txt

*YSR Script Files*

rs\_AssocLetter\_YSR.txt

CndLtr

*YSR Template Names*

YSR\_AssocLetter.xlsx

CndLtr

**Correpondence Condo Purchase (CondoPurchase)***Top-Level Select Script Files*

rs\_AssocPurchase\_YSR.txt

*YSR Script Files*

rs\_AssocPurchase\_YSR.txt

purchase

*YSR Template Names*

yDoc\_CondoPurchase.doc

purchase

**RentCafeCondo Work Orders Report (RentCafeCondoWO)***Top-Level Select Script Files**YSR Script Files*

YSR\_CondoWorkOrders.txt

CondoWO

*YSR Template Names*

YSR\_CondoWorkOrders.xlsx

CondoWO

## Construction

**Budget Detail (BudgetDetail)***Top-Level Select Script Files**YSR Script Files*

rs\_Const\_YSR\_BudgetDet.txt

R1

*YSR Template Names*

rs\_Const\_YSR\_BudgetDet.xlsx

R1

#### **Percent Complete Invoice (PCInv)**

*Top-Level Select Script Files*

*YSR Script Files*

rs\_Const\_YSR\_PCInvoice.txt

R1

rs\_Const\_YSR\_PCInvoice\_Detail.txt

R2

*YSR Template Names*

rs\_Const\_YSR\_PCInvoice.xlsx

R1

rs\_Const\_YSR\_PCInvoice\_Detail.xlsx

R2

#### **Project Profitability Report (ProjectProfitRpt)**

*Top-Level Select Script Files*

*YSR Script Files*

rs\_YSR\_ProjectProfitability.txt

Main

*YSR Template Names*

rs\_YSR\_ProjectProfitability.xlsx

Main

#### **Request for Payment (ReqPmt)**

*Top-Level Select Script Files*

*YSR Script Files*

rs\_Const\_YSR\_RequestPmt.txt

R1

rs\_Const\_YSR\_RequestPmt\_Detail.txt

R2

*YSR Template Names*

rs\_Const\_YSR\_RequestPmt.xlsx

R1

rs\_Const\_YSR\_RequestPmt\_Detail.xlsx

R2

## **DAS**

#### **DAS Approval/Rejection Report (DASAR)**

*Top-Level Select Script Files*

*YSR Script Files*

rs\_sql\_DasReject.txt

DASAR

*YSR Template Names*

DASRejectReport.xlsx

DASAR

## **Financial Analytics**

#### **Aging Detail Report (AginDet)**

*Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR\_ARAgingDetail.xlsx

AgingDet

**AR Aging Summary Report (AgingSum)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR\_ARAgingSummary.xlsx

AgingSum

**Balance Sheet (BalanceS)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR\_BalanceSheet.xlsx

BalanceS

**Budget Comparison (BudgetCo)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR\_BudgetComparison.xlsx

BudgetCo

**Charge Register (ChargReg)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR\_TransRegister\_Charge.xlsx

ChargReg

**Credit note - Sample Word Template (CreditN)***Top-Level Select Script Files**YSR Script Files*

rs\_YSR\_CreditNote.txt

credit

*YSR Template Names*

rs\_YSR\_CreditNote.docx

credit

**Custom Financial Income Statement (CF\_IS)***Top-Level Select Script Files*

YSR\_Custom\_Financial\_Income\_Statement.txt

<i>YSR Script Files</i>		
YSR_Custom_Financial_Income_Statement.txt		CFIS
<i>YSR Template Names</i>		
YSR_Custom_Financial_Income_Statement.xlsx		CFIS
<b>Income Statement (IncomeSt)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
YSR_IncomeStatement.xlsx		IncomeSt
<b>Invoice Register (InvoiceR)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
YSR_TransRegister_Invoice.xlsx		InvoiceR
<b>Payable Register (PayableR)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
YSR_TransRegister_Payable.xlsx		PayableR
<b>Receipt Register (Receipt)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
YSR_TransRegister_Receipt.xlsx		Receipt
<b>Trial Balance (TrialBal)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
YSR_TrialBalance.xlsx		TrialBal

## Genesis Two

### Debtors Listing with Notes (DebtorsListing)

*Top-Level Select Script Files*

*YSR Script Files*

rs_sql_Debtors_Listing_with_Notes.txt	Debtors
---------------------------------------	---------

*YSR Template Names*

SELECT CASE '#subtotals#' WHEN 'Yes' THEN 'YSR_Debtors_Listing_with_Notes_Subtotals.xlsxm' ELSE 'YSR_Debtors_Listing_with_Notes.xlsxm' END	Debtors
---	---------

**Lease Abstract Report (LeaseAbstract)***Top-Level Select Script Files*

YSR_rs_G2_Comm_LeaseAbstract.txt	LeaseAbt
----------------------------------	----------

*YSR Script Files*

YSR_rs_G2_Comm_LeaseAbstract.txt	LeaseAbt
----------------------------------	----------

*YSR Template Names*

YSR_Custom_Lease_Abstract.doc	LeaseAbt
-------------------------------	----------

**Owner Monthly Report (OwnerPac)***Top-Level Select Script Files*

<i>YSR Script Files</i>	
G2_Cash_Flow.txt	CashFlw
G2_Expense_Reg.txt	ExpReg
G2_Owner_CoverLetter.txt	GenLed
G2_Owner_CoverLetter.txt	IncomeSt
G2_Owner_CoverLetter.txt	JournalR
G2_Owner_CoverLetter.txt	PayableR
G2_Owner_CoverLetter.txt	ReceiptR
G2_Owner_CoverLetter.txt	RentRoll
G2_Owner_CoverLetter.txt	ChargeRe
G2_Owner_CoverLetter.txt	

## CheckReg

G2_Owner_CoverLetter.txt	CoverPag
G2_Owner_CoverLetter.txt	

## ExpenseD

G2_Owner_CoverLetter.txt	AgingDet
G2_Owner_CoverLetter.txt	

## AgingSum

G2_Owner_CoverLetter.txt	APAging
G2_Owner_CoverLetter.txt	BalanceS
G2_Owner_CoverLetter.txt	

## BudgetCo

G2_Owner_CoverLetter.txt	CashFlow
G2_Owner_CoverLetter.txt	TrialBal
G2_Owner_CoverLetter.txt	WorkOrd
G2_Owner_Statement.txt	

OwnrStmt		
G2_Res_RentRoll_Owner.txt	RRentRol	
G2_Transactions_Reg.txt	TranReg	
<i>YSR Template Names</i>		
G2_ARAgingDetail.xlsx	AgingDet	
G2_ARAgingSummary.xlsx		
AgingSum		
G2_APAGing.xlsx	APAGing	
G2_BalanceSheet.xlsx	BalanceS	
G2_BudgetComparison.xlsx		
BudgetCo		
G2_CashFlow.xlsx	CashFlow	
G2_Cash_Flow.xlsx	CashFlw	
G2_TransRegister_Charge.xlsx	ChargeRe	
G2_TransRegister_Check.xlsx		
CheckReg		
G2_Owner_CoverPage.xlsx	CoverPag	
G2_APExpenseDistribution.xlsx		
ExpenseD		
G2_ExpenseReg.xlsx	ExpReg	
G2_General_Ledger.xlsx	GenLed	
G2_IncomeStatement.xlsx	IncomeSt	
G2_TransRegister_Journal.xlsx	JournalR	
G2_OwnerStatement.xlsx		
OwnrStmt		
G2_TransRegister_Payable.xlsx	PayableR	
G2_TransRegister_Receipt.xlsx	ReceiptR	
G2_RentRoll.xlsx	RentRoll	
G2_Res_RentRoll.xlsx	RRentRol	
G2_TransRegister.xlsx	TranReg	
G2_TrialBalance.xlsx	TrialBal	
G2_WorkOrder.xlsx	WorkOrd	

**Owner Monthly Report - 1 Tax (Ownr1Tax)***Top-Level Select Script Files**YSR Script Files*

G2_Cash_Flow.txt	CashFlw
G2_Expense_Reg.txt	ExpReg
G2_Owner_CoverLetter.txt	GenLed
G2_Owner_CoverLetter.txt	IncomeSt
G2_Owner_CoverLetter.txt	JournalR
G2_Owner_CoverLetter.txt	PayableR
G2_Owner_CoverLetter.txt	ReceiptR

G2_Owner_CoverLetter.txt	RentRoll
G2_Owner_CoverLetter.txt	ChargeRe
G2_Owner_CoverLetter.txt	
CheckReg	
G2_Owner_CoverLetter.txt	CoverPag
G2_Owner_CoverLetter.txt	
ExpenseD	
G2_Owner_CoverLetter.txt	AgingDet
G2_Owner_CoverLetter.txt	
AgingSum	
G2_Owner_CoverLetter.txt	APAging
G2_Owner_CoverLetter.txt	BalanceS
G2_Owner_CoverLetter.txt	
BudgetCo	
G2_Owner_CoverLetter.txt	CashFlow
G2_Owner_CoverLetter.txt	TrialBal
G2_Owner_CoverLetter.txt	WorkOrd
G2_Owner_Statement.txt	
OwnrStmt	
G2_Res_RentRoll_Owner.txt	RRentRol
G2_Transactions_Reg.txt	TranReg
<i>YSR Template Names</i>	
G2_ARAgingDetail_Tax.xlsx	AgingDet
G2_ARAgingSummary.xlsx	
AgingSum	
G2_APAGing_Tax.xlsx	APAging
G2_BalanceSheet.xlsx	BalanceS
G2_BudgetComparison.xlsx	
BudgetCo	
G2_CashFlow.xlsx	CashFlow
G2_Cash_Flow.xlsx	CashFlw
G2_TransRegister_Charge_1Tax.xlsx	ChargeRe
G2_TransRegister_Check_1Tax.xlsx	
CheckReg	
G2_Owner_CoverPage.xlsx	CoverPag
G2_APExpenseDistribution.xlsx	
ExpenseD	
G2_ExpenseReg.xlsx	ExpReg
G2_General_Ledger.xlsx	GenLed
G2_IncomeStatement.xlsx	IncomeSt
G2_TransRegister_Journal.xlsx	JournalR
G2_OwnerStatement.xlsx	
OwnrStmt	

G2_TransRegister_Payable_1Tax.xlsx	PayableR
G2_TransRegister_Receipt_1Tax.xlsx	ReceiptR
G2_RentRoll.xlsx	RentRoll
G2_Res_RentRoll.xlsx	RRentRol
G2_TransRegister.xlsx	TranReg
G2_TrialBalance.xlsx	TrialBal
G2_WorkOrder.xlsx	WorkOrd

**Owner Monthly Report - 2 Taxes (Ownr2Tax)***Top-Level Select Script Files**YSR Script Files*

G2_Cash_Flow.txt	CashFlw
G2_Expense_Reg.txt	ExpReg
G2_Owner_CoverLetter.txt	GenLed
G2_Owner_CoverLetter.txt	IncomeSt
G2_Owner_CoverLetter.txt	JournalR
G2_Owner_CoverLetter.txt	PayableR
G2_Owner_CoverLetter.txt	ReceiptR
G2_Owner_CoverLetter.txt	RentRoll
G2_Owner_CoverLetter.txt	ChargeRe
CheckReg	
G2_Owner_CoverLetter.txt	CoverPag
G2_Owner_CoverLetter.txt	
ExpenseD	
G2_Owner_CoverLetter.txt	AgingDet
G2_Owner_CoverLetter.txt	
AgingSum	
G2_Owner_CoverLetter.txt	APAging
G2_Owner_CoverLetter.txt	BalanceS
G2_Owner_CoverLetter.txt	
BudgetCo	
G2_Owner_CoverLetter.txt	CashFlow
G2_Owner_CoverLetter.txt	TrialBal
G2_Owner_CoverLetter.txt	WorkOrd
G2_Owner_Statement.txt	
OwnrStmt	
G2_Res_RentRoll_Owner.txt	RRentRol
G2_Transactions_Reg.txt	TranReg

*YSR Template Names*

G2_ARAgingDetail_Tax.xlsx	AgingDet
G2_ARAgingSummary.xlsx	

AgingSum

G2_APAGing_Tax.xlsx	APAGing
G2_BalanceSheet.xlsx	BalanceS
G2_BudgetComparison.xlsx	
BudgetCo	
G2_CashFlow.xlsx	CashFlow
G2_Cash_Flow.xlsx	CashFlw
G2_TransRegister_Charge_2Tax.xlsx	ChargeRe
G2_TransRegister_Check_2Tax.xlsx	
CheckReg	
G2_Owner_CoverPage.xlsx	CoverPag
G2_APExpenseDistribution.xlsx	
ExpenseD	
G2_ExpenseReg.xlsx	ExpReg
G2_General_Ledger.xlsx	GenLed
G2_IncomeStatement.xlsx	IncomeSt
G2_TransRegister_Journal.xlsx	JournalR
G2_OwnerStatement.xlsx	
OwnrStmt	
G2_TransRegister_Payable_2Tax.xlsx	PayableR
G2_TransRegister_Receipt_2Tax.xlsx	ReceiptR
G2_RentRoll.xlsx	RentRoll
G2_Res_RentRoll.xlsx	RRentRol
G2_TransRegister.xlsx	TranReg
G2_TrialBalance.xlsx	TrialBal
G2_WorkOrder.xlsx	WorkOrd

**Property Report Packet (PropPack)***Top-Level Select Script Files**YSR Script Files*

G2_Cash_Flow_Property.txt	CashFlw
G2_Property_CoverLetter.txt	ChargeRe
G2_Property_CoverLetter.txt	

## CheckReg

G2_Property_CoverLetter.txt	CoverPag
G2_Property_CoverLetter.txt	

## ExpenseD

G2_Property_CoverLetter.txt	GenLed
G2_Property_CoverLetter.txt	IncomeSt
G2_Property_CoverLetter.txt	JournalR
G2_Property_CoverLetter.txt	PayableR
G2_Property_CoverLetter.txt	ReceiptR
G2_Property_CoverLetter.txt	RentRoll
G2_Property_CoverLetter.txt	AgingDet
G2_Property_CoverLetter.txt	

AgingSum	G2_Property_CoverLetter.txt	APAgings
	G2_Property_CoverLetter.txt	BalanceS
	G2_Property_CoverLetter.txt	
BudgetCo	G2_Property_CoverLetter.txt	CashFlow
	G2_Property_CoverLetter.txt	TrialBal
	G2_Property_CoverLetter.txt	WorkOrd
	G2_Res_RentRoll.txt	RRentRol
<i>YSR Template Names</i>		
	G2_ARAgingDetail.xlsx	AgingDet
	G2_ARAgingSummary.xlsx	
AgingSum	G2_APAgings.xlsx	APAgings
	G2_BalanceSheet.xlsx	BalanceS
	G2_BudgetComparison.xlsx	
BudgetCo	G2_CashFlow.xlsx	CashFlow
	G2_Cash_Flow.xlsx	CashFlw
	G2_TransRegister_Charge.xlsx	ChargeRe
	G2_TransRegister_Check.xlsx	
CheckReg	G2_Property_CoverPage.xlsx	CoverPag
	G2_APExpenseDistribution.xlsx	
ExpenseD	G2_General_Ledger.xlsx	GenLed
	G2_IncomeStatement.xlsx	IncomeSt
	G2_TransRegister_Journal.xlsx	JournalR
	G2_TransRegister_Payable.xlsx	PayableR
	G2_TransRegister_Receipt.xlsx	ReceiptR
	G2_RentRoll.xlsx	RentRoll
	G2_Res_RentRoll.xlsx	RRentRol
	G2_TrialBalance.xlsx	TrialBal
	G2_WorkOrder.xlsx	WorkOrd

**Property Report Packet - 1 Tax (Prop1Tax)***Top-Level Select Script Files**YSR Script Files*

G2_Cash_Flow_Property.txt	CashFlw
G2_Property_CoverLetter.txt	ChargeRe
G2_Property_CoverLetter.txt	

CheckReg

G2_Property_CoverLetter.txt	CoverPag
-----------------------------	----------

G2_Property_CoverLetter.txt	
ExpenseD	
G2_Property_CoverLetter.txt	GenLed
G2_Property_CoverLetter.txt	IncomeSt
G2_Property_CoverLetter.txt	JournalR
G2_Property_CoverLetter.txt	PayableR
G2_Property_CoverLetter.txt	ReceiptR
G2_Property_CoverLetter.txt	RentRoll
G2_Property_CoverLetter.txt	AgingDet
AgingSum	
G2_Property_CoverLetter.txt	APAging
G2_Property_CoverLetter.txt	Balances
G2_Property_CoverLetter.txt	
BudgetCo	
G2_Property_CoverLetter.txt	CashFlow
G2_Property_CoverLetter.txt	TrialBal
G2_Property_CoverLetter.txt	WorkOrd
G2_Res_RentRoll.txt	RRentRol
<i>YSR Template Names</i>	
G2_ARAgingDetail_Tax.xlsx	AgingDet
G2_ARAgingSummary.xlsx	
AgingSum	
G2_APAGing_Tax.xlsx	APAging
G2_BalanceSheet.xlsx	Balances
G2_BudgetComparison.xlsx	
BudgetCo	
G2_CashFlow.xlsx	CashFlow
G2_Cash_Flow.xlsx	CashFlw
G2_TransRegister_Charge_1Tax.xlsx	ChargeRe
G2_TransRegister_Check_1Tax.xlsx	
CheckReg	
G2_Property_CoverPage.xlsx	CoverPag
G2_APExpenseDistribution.xlsx	
ExpenseD	
G2_General_Ledger.xlsx	GenLed
G2_IncomeStatement.xlsx	IncomeSt
G2_TransRegister_Journal.xlsx	JournalR
G2_TransRegister_Payable_1Tax.xlsx	PayableR
G2_TransRegister_Receipt_1Tax.xlsx	ReceiptR
G2_RentRoll.xlsx	RentRoll
G2_Res_RentRoll.xlsx	RRentRol
G2_TrialBalance.xlsx	TrialBal

G2_WorkOrder.xlsx	WorkOrd
<b>Property Report Packet - 2 Taxes (Prop2Tax)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
G2_Cash_Flow_Property.txt	CashFlw
G2_Property_CoverLetter.txt	ChargeRe
G2_Property_CoverLetter.txt	
CheckReg	
G2_Property_CoverLetter.txt	CoverPag
G2_Property_CoverLetter.txt	
ExpenseD	
G2_Property_CoverLetter.txt	GenLed
G2_Property_CoverLetter.txt	IncomeSt
G2_Property_CoverLetter.txt	JournalR
G2_Property_CoverLetter.txt	PayableR
G2_Property_CoverLetter.txt	ReceiptR
G2_Property_CoverLetter.txt	RentRoll
G2_Property_CoverLetter.txt	AgingDet
G2_Property_CoverLetter.txt	
AgingSum	
G2_Property_CoverLetter.txt	APAging
G2_Property_CoverLetter.txt	BalanceS
G2_Property_CoverLetter.txt	
BudgetCo	
G2_Property_CoverLetter.txt	CashFlow
G2_Property_CoverLetter.txt	TrialBal
G2_Property_CoverLetter.txt	WorkOrd
G2_Res_RentRoll.txt	RRentRol
<i>YSR Template Names</i>	
G2_ARAgingDetail_Tax.xlsx	AgingDet
G2_ARAgingSummary.xlsx	
AgingSum	
G2_APAGing_Tax.xlsx	APAging
G2_BalanceSheet.xlsx	BalanceS
G2_BudgetComparison.xlsx	
BudgetCo	
G2_CashFlow.xlsx	CashFlow
G2_Cash_Flow.xlsx	CashFlw
G2_TransRegister_Charge_2Tax.xlsx	ChargeRe
G2_TransRegister_Check_2Tax.xlsx	
CheckReg	
G2_Property_CoverPage.xlsx	CoverPag

G2_APExpenseDistribution.xlsx	
ExpenseD	
G2_General_Ledger.xlsx	GenLed
G2_IncomeStatement.xlsx	IncomeSt
G2_TransRegister_Journal.xlsx	JournalR
G2_TransRegister_Payable_2Tax.xlsx	PayableR
G2_TransRegister_Receipt_2Tax.xlsx	ReceiptR
G2_RentRoll.xlsx	RentRoll
G2_Res_RentRoll.xlsx	RRentRol
G2_TrialBalance.xlsx	TrialBal
G2_WorkOrder.xlsx	WorkOrd

**Tenant Ledger (TenantLdgr)***Top-Level Select Script Files*

YSR\_G2\_Tenant\_Ledger.txt

*YSR Script Files*

YSR\_G2\_Tenant\_Ledger.txt

TenantLd

*YSR Template Names*

YSR\_G2\_Tenant\_Ledger.xlsx

TenantLd

**Tenant Statement (TenantStmt)***Top-Level Select Script Files*

YSR\_rs\_G2\_Tenant\_Statement.SSRS.txt

*YSR Script Files*

YSR\_rs\_G2\_Tenant\_Statement.SSRS.txt

TenantSt

*YSR Template Names*

YSR\_G2\_Tenant\_Statement.xlsx

TenantSt

## Government

**Gov Coast Guard Report of Certificates (GovRegCert)***Top-Level Select Script Files**YSR Script Files*

rs\_Gov\_Vessel\_YSR\_RegistrationReportofCertificates.txt

GovRegTe

*YSR Template Names*

YSR\_Gov\_Vessel\_RegistrationReportofCertificates.xlsx

GovRegTe

**Gov Coast Guard State Registration Data (GovRegData)***Top-Level Select Script Files*

*YSR Script Files*

rs\_Gov\_Vessel\_YSR\_StateRegistrationData.txt

GovRegTe

*YSR Template Names*

YSR\_Gov\_Vessel\_StateRegistrationData.xlsx

GovRegTe

**Gov Commercial Operation Detail (GovRegDetail)***Top-Level Select Script Files**YSR Script Files*

rs\_Gov\_Vessel\_YSR\_CommercialOperationDetail.txt

GovRegDe

*YSR Template Names*

YSR\_Gov\_Vessel\_CommercialOperationDetail.xlsx

GovRegDe

**Gov Commercial Operation Summary (GovRegCo)***Top-Level Select Script Files**YSR Script Files*

rs\_Gov\_Vessel\_YSR\_CommercialOperationSummary.txt

GovRegCo

*YSR Template Names*

YSR\_Gov\_Vessel\_CommercialOperationSummary.xlsx

GovRegCo

**Gov Registration Percentage Summary (GovRegSumm)***Top-Level Select Script Files**YSR Script Files*

rs\_Gov\_Vessel\_YSR\_RegistrationPercentageSummary.txt

GovRegTe

*YSR Template Names*

YSR\_Gov\_Vessel\_RegistrationPercentageSummary.xlsx

GovRegTe

**Gov Registration Summary by Length (GovRegSumLen)***Top-Level Select Script Files**YSR Script Files*

rs\_Gov\_Vessel\_YSR\_RegistrationSummarybyLength.txt

GovRegTe

*YSR Template Names*

YSR\_Gov\_Vessel\_RegistrationSummarybyLength.xlsx

GovRegTe

### **Gov Vessel Registration (GovReg)**

*Top-Level Select Script Files*

rs\_Gov\_Vessel\_YSR\_RegistrationLetters.txt

*YSR Script Files*

rs\_Gov\_Vessel\_YSR\_RegistrationLetters.txt

GovReg

*YSR Template Names*

```
select case when '#PrePrinted#' = 'Yes' then 'YSR_GovVessel_PrePrintedRegistrationLetter.docx'  
else 'YSR_GovVessel_RegistrationLetter.docx' end
```

GovReg

### **Gov Vessel Temporary Registration (GovTempReg)**

*Top-Level Select Script Files*

rs\_Gov\_Vessel\_YSR\_TemporaryRegistrationLetters.txt

*YSR Script Files*

rs\_Gov\_Vessel\_YSR\_TemporaryRegistrationLetters.txt

GovTempR

*YSR Template Names*

```
select case when '#PrePrinted#' = 'Yes' then
```

GovTempR

```
'YSR_GovVessel_TemporaryPrePrintedRegistrationLetter.docx' else
```

```
'YSR_GovVessel_TemporaryRegistrationLetter.docx' end
```

## **Inspections**

### **Inspection - PHA - HQS Report Failed Items Only (INSPPHAHQSFail)**

*Top-Level Select Script Files*

rs\_Insp2\_PHA\_YSR\_HQS\_Failed\_only.txt

*YSR Script Files*

rs\_Insp2\_PHA\_YSR\_HQS\_Failed\_only.txt

InspHQS

*YSR Template Names*

```
Select 'YSR_Insp2_PHA_HQS_Failed_Only_' + case when '#RptType#'='Summary List' then  
'SummaryList' else 'Letter' end + '.docx'
```

InspHQS

### **Inspection Details Report (INSPDetail)**

*Top-Level Select Script Files*

*YSR Script Files*

*YSR Template Names*

```
Select Case when '#PrintPhotos#'='Yes' then 'YSR_Insp2_InspectionDetailsReportPhoto.xlsx'  
else 'YSR_Insp2_InspectionDetailsReport.xlsx' end
```

InspDetH

**Inspection Duration Report (INSPDuration)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR\_Insp2\_InspectionDuration.xlsx

inspDura

**Inspection Letters (INSPLetter)***Top-Level Select Script Files*

rs\_Insp2\_YSR\_Letters.txt

*YSR Script Files*

rs\_Insp2\_YSR\_Letters.txt

InspLet

*YSR Template Names*

Select 'YSR\_Insp2\_'+REPLACE('#RptType#','-', '')+'Letter.docx'

InspLet

**Inspection Letters PHA (INSPPHALetter)***Top-Level Select Script Files*

rs\_Insp2\_PHA\_YSR\_InspLetters.txt

*YSR Script Files*

rs\_Insp2\_PHA\_YSR\_InspLetters.txt

InspLet

*YSR Template Names*

Select 'YSR\_Insp2\_PHA\_InspLetters\_ '+REPLACE(REPLACE('#RptType#',' - ','\_'), 'Schedule', 'Scheduled')+'.docx'

InspLet

**Inspection Letters PHA (Inspection Screen) (INSPPHARHSLetter)***Top-Level Select Script Files**YSR Script Files*

rs\_Insp2\_PHA\_YSR\_InspLetters\_1.txt

PHARHS

rs\_Insp2\_PHA\_YSR\_InspLetters\_FailedDetails\_1.txt

Failed

*YSR Template Names*select Case when '#RptType#='Result' then 'YSR\_Insp2\_PHA\_InspLetters\_Failed\_Items.docx'  
else null end

Failed

SELECT 'YSR\_Insp2\_PHA\_InspLetters\_ '+ case when '#RptType#='Schedule' then'Scheduled'  
when '#RptType#='Result' then Case when I.sStatus in ('Pass','Fail','Inconclusive','Cancel') then  
replace(I.sStatus,'Cancel','Canceled') else 'Inconclusive'end + Case when '#Notes#'='Yes' then  
'\_Notes' else '' end Else 'Inconclusive\_Notes'end+'.docx' from Inspection2 I where I.hMy  
=#InspectionID#

PHARHS

**Inspection Summary Report (INSPSummary)***Top-Level Select Script Files**YSR Script Files*

*YSR Template Names*

YSR_Insp2_InspectionSummary_Asset.xlsx	Asset
YSR_Insp2_InspectionSummary_Building.xlsx	Building
YSR_Insp2_InspectionSummary_DueDiligenceProperty.xlsx	DDProp
YSR_Insp2_InspectionSummary_DueDiligenceUnit.xlsx	DDUnit
YSR_Insp2_InspectionSummary_Property.xlsx	Property
YSR_Insp2_InspectionSummary_Room.xlsx	Room
YSR_Insp2_InspectionSummary_Unit.xlsx	Unit

**Inspection Unassigned Directory Report (INSPUnassigned)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR_Insp2_InspectionUnassignedDirectory.xlsx	InspUnas
--	----------

**Inspection Uncompleted Date Review Report (INSPUncompleted)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR_Insp2_InspectionUncompleted.xlsx	InspUnco
--------------------------------------	----------

## International

**Bas Report APAC Generic (Bas2)***Top-Level Select Script Files*

rs\_Int60\_BAS\_Summary.txt

*YSR Script Files*

rs_Int60_BAS_Summary.txt	BAS
--------------------------	-----

*YSR Template Names*

select case '#reporttype#' when 'Summary' then 'rs_Int60_BAS_Summary.xlsx' else 'rs_Int60_BAS_Detail.xlsx' end	BAS
---	-----

**Bas Report APAC Taxpoint (Bas2Tx)***Top-Level Select Script Files*

rs\_Int60\_BAS2Tx\_Summary.txt

*YSR Script Files*

rs_Int60_BAS2Tx_Summary.txt	BAS
-----------------------------	-----

*YSR Template Names*

select case '#reporttype#' when 'Summary' then 'rs_Int60_BAS2Tx_Summary.xlsx' else 'rs_Int60_BAS2Tx_Detail.xlsx' end	BAS
---	-----

**Correspondence International Invoice (CMIInv)***Top-Level Select Script Files*

ss\_YSR\_InternationalInvoice.txt

*YSR Script Files*

ss\_YSR\_InternationalInvoice.txt

CMinv

*YSR Template Names*

YSR\_InternationalInvoice.xlsx

CMinv

**General Client Statement (GenCS)***Top-Level Select Script Files*

rs\_YSR\_CS\_OwnerListing.txt

*YSR Script Files*

rs\_YSR\_CS\_1\_CoverPage.txt

StmtInfo

rs\_YSR\_CS\_2\_StatementSummary.txt

StmtSumm

rs\_YSR\_CS\_3\_IncomeStatement.txt

IncStmt

rs\_YSR\_CS\_4\_ExpenseStatement.txt

ExpStmt

rs\_YSR\_CS\_5\_CashMovement.txt

Movement

*YSR Template Names*

YSR\_CS\_4\_ExpenseStatement.xlsx

ExpStmt

YSR\_CS\_3\_IncomeStatement.xlsx

IncStmt

YSR\_CS\_5\_CashMovement.xlsx

Movement

YSR\_CS\_1\_CoverPage.xlsx

StmtInfo

YSR\_CS\_2\_StatementSummary.xlsx

StmtSumm

**International Residential Lease Document (LeaseDoc)***Top-Level Select Script Files**YSR Script Files*

rs\_YSR\_Intres\_LeaseDoc.txt

L1

*YSR Template Names*

YSR\_Intres\_LeaseDoc.xlsx

L1

**International Residential Lease Expiration (IntResLeaseExp)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR\_IntRes\_LeaseExpiration.xlsx

IRLE

**International Residential Market Rent Schedule**

*Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR\_IntRes\_MarketRentSchedule.xlsx

IRMRS

**International Residential PDC (PDC)***Top-Level Select Script Files**YSR Script Files*

rs\_YSR\_Intres\_PDC.txt

P1

*YSR Template Names*

YSR\_Intres\_PDC.xlsx

P1

**International Residential Potential Rent (IntResPotRent)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR\_IntRes\_PotentialRent.xlsx

IRPR

**International Residential Unit Availability (IntResUnitAva)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR\_IntRes\_UnitAvailability.xlsx

IRUA

**NL Annual Rent Increase (NLRentInc)***Top-Level Select Script Files*

rs\_ysr\_NLRentInc.txt

*YSR Script Files*

rs\_ysr\_NLRentInc.txt

NLRent

*YSR Template Names*

rs\_ysr\_NLRentInc.xlsx

NLRent

**NL scoring details report (NLscoringRpt)***Top-Level Select Script Files**YSR Script Files*

rs\_ysr\_NLScoringInfo.txt

NLInfo

*YSR Template Names*

rs\_ysr\_NLScoringInfo.xlsx

NLInfo

**Tenant Ledger International (INT\_TenantLedger)**

*Top-Level Select Script Files**YSR Script Files*

rs\_sql\_Tenant\_Ledger\_INT.TXT

Report

*YSR Template Names*

YSR\_Tenant\_Ledger\_INT.xlsx

Report

## International MEAA

**Custom Purchase Order (PO)***Top-Level Select Script Files*

rs\_APAC\_PurchaseOrder.txt

*YSR Script Files*

rs\_APAC\_PurchaseOrder.txt

po

*YSR Template Names*

ydoc\_APAC\_PurchaseOrder.docx

po

Temp.xlsx

po2

**Custom Work Order (WO)***Top-Level Select Script Files*

rs\_APAC\_WorkOrder.txt

*YSR Script Files*

rs\_APAC\_WorkOrder.txt

wo

*YSR Template Names*

ydoc\_APAC\_WorkOrder.docx

wo

temp.xlsx

wo2

**New Owner Statement (Ostmt)***Top-Level Select Script Files*

rs\_MEAA\_Owner\_Statement.txt

*YSR Script Files*

rs\_MEAA\_Owner\_Statement.txt

InvImg

rs\_MEAA\_Owner\_Statement.txt

OwnerRep

*YSR Template Names*

YSR\_MEAA\_Owner\_Statement\_yDoc.docx

Dummy

YSR\_MEAA\_Owner\_Statement\_Invoices.xlsx

InvImg

YSR\_MEAA\_Owner\_Statement.xlsx

OwnerRep

**Report Packet (RprtPkt)***Top-Level Select Script Files*

*YSR Script Files*

rs_APAC_RprtPkt_ARPayment.txt	ARP
rs_APAC_RprtPkt_ARPayment.txt	ARPD
rs_APAC_RprtPkt_ARPayment.txt	ARPDG
rs_APAC_RprtPkt_BalanceSheet.txt	BalSh
rs_APAC_RprtPkt_BAS_Detail.txt	BASDet
rs_APAC_RprtPkt_BAS_Summary.txt	BASSum
rs_APAC_RprtPkt_IncExpStmt.txt	IncExp
rs_APAC_RprtPkt_IncExpStmt.txt	IncExpG
rs_APAC_RprtPkt_IncomeStatement.txt	IncStmt
rs_APAC_RprtPkt_Index_Page.txt	Index
rs_APAC_RprtPkt_MemoListing.txt	Memo1
rs_apac_RprtPkt_Owner_Statement.txt	OStmt
rs_apac_RprtPkt_Owner_Statement.txt	OStmtImg
rs_APAC_RprtPkt_Receivable_Ledger.txt	RecLegr
rs_APAC_RprtPkt_Tenancy_Schedule2.txt	CustTS2
rs_APAC_RprtPkt_WALE_ByProperty_MLA.txt	Wale
rs_APAC_RprtPkt_YSR_APAnalytics.txt	APExpD
rs_APAC_RprtPkt_YSR_ARAgingDetail.txt	ARAgS
rs_APAC_RprtPkt_YSR_ARAgingDetail.txt	AgingDet
rs_APAC_RprtPkt_YSR_ARAgingDetail.txt	AgSG
rs_APAC_RprtPkt_YSR_InternationalAcctLedger.txt	AccLedgr
rs_APAC_RprtPkt_YSR_TrialBalance.txt	TrialBal

*YSR Template Names*

rx_APAC_RprtPkt_InternationalAcctLedger.xls	AccLedgr
rx_APAC_RprtPkt_ARAgingDetail.xls	AgingDet
rx_APAC_RprtPkt_ARAgingSummary_Graph.xls	AgSG
rx_APAC_RprtPkt_APExpenseDistribution.xls	APExpD
rx_APAC_RprtPkt_ARAgingSummary.xls	ARAgS
rx_APAC_RprtPkt_ARPayment_Summary.xls	ARP
rx_APAC_RprtPkt_ARPayment_Detail.xls	ARPD
rx_APAC_RprtPkt_ARPayment_Detail_Graph.xls	ARPDG
rx_APAC_RprtPkt_BalanceSheet.xls	BalSh
rx_APAC_RprtPkt_BAS_Detail.xls	BASDet
rx_APAC_RprtPkt_BAS_Summary.xls	BASSum
rx_APAC_RprtPkt_Tenancy_Schedule2.xls	CustTS2
rx_APAC_RprtPkt_IncExpStmt.xls	IncExp
rx_APAC_RprtPkt_IncExpStmtGraph.xls	IncExpG
rx_APAC_RprtPkt_IncomeStatement.xls	IncStmt
rx_APAC_RprtPkt_Index_Page.xls	Index
rx_APAC_RprtPkt_MemoListing.xls	Memo1
rx_APAC_RprtPkt_Owner_Statement.xls	OStmt

rx\_APAC\_RprtPkt\_Owner\_Statement\_Img.xls  
 OSmtlImg  
 rx\_APAC\_RprtPkt\_Receivable\_Ledger.xls  
 rx\_APAC\_RprtPkt\_YSR\_TrialBalance.xls  
 rx\_APAC\_RprtPkt\_ydoc\_WALE\_ByProperty.xls

RecLegr  
 TrialBal  
 Wale

### Tax Invoice (TaxInv)

*Top-Level Select Script Files*

rs\_APAC\_YSR\_Tax\_Invoice.txt

*YSR Script Files*

rs\_APAC\_YSR\_Tax\_Invoice.txt  
 rs\_APAC\_YSR\_Tax\_Invoice.txt  
 rs\_APAC\_YSR\_Tax\_Invoice.txt

dummy  
 Inv  
 InvImg

*YSR Template Names*

dummy.xlsx  
 rs\_APAC\_YSR\_Tax\_Invoice.docx  
 rs\_APAC\_ydoc\_Payable\_Img.docx

dummy  
 Inv  
 InvImg

### Tenancy Schedule (TenancySchedule)

*Top-Level Select Script Files*

rs\_APAC\_TenancySchedule\_Header.txt

*YSR Script Files*

rs\_APAC\_TenancySchedule.txt

main

*YSR Template Names*

SELECT CASE WHEN '#showPropertySummary#' = 1 THEN  
 'rx\_APAC\_TenancySchedule\_Summary.xlsx' ELSE 'rx\_APAC\_TenancySchedule.xlsx' END

main

### Trust Reconciliation Report (TrustRec)

*Top-Level Select Script Files*

rs\_APAC\_TrustRecRep.txt

*YSR Script Files*

rs\_APAC\_TrustRecRep.txt

main

*YSR Template Names*

rs\_APAC\_TrustRecRep.xlsx

main

## Investment Management

### Capital Balance Report Investment View (cbal\_inv)

*Top-Level Select Script Files*

*YSR Script Files*

<i>YSR Template Names</i>		
capbal_investment.xlsx		capbal11
<b>Capital Balance Report Investor View (cap_bal)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
capbal_data.xlsx		capbal
<b>Capital Call Report (capcallre)</b>		
<i>Top-Level Select Script Files</i>		
rs_capcall_report_header.txt		
<i>YSR Script Files</i>		
rs_capcall_report_detail.txt		capcall
<i>YSR Template Names</i>		
rs_capitalcall_report.xlsx		capcall
<b>Capital Call Report Single Currency (singcpcl)</b>		
<i>Top-Level Select Script Files</i>		
rs_capcall_singlecurr_report_header.txt		
<i>YSR Script Files</i>		
rs_capcall_singlecurr_report_detail.txt		capcall
<i>YSR Template Names</i>		
rs_capitalcall_singlecurr_report.xlsx		capcall
<b>Custom IM Investment Attribute Report (custIM_invsatt)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
custim_invstmntattr.xlsx		cusim
<b>Custom IM Investment Holding Report (CustIM_Invstmtnt)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
customim_investment.xlsx		custim
<b>Custom IM Investor Attribute Report (cusIM_invattr)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		

<i>YSR Template Names</i>		
customim_investorattr.xlsx		custIM
<b>Custom IM Investor Holding Report (CustIM_invstr)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
customim_investor.xlsx		custIM
<b>Distribution Notice (dstnotic)</b>		
<i>Top-Level Select Script Files</i>		
rs_distnotice_header.txt		
<i>YSR Script Files</i>		
rs_distnotice_detail.txt		dstnot
<i>YSR Template Names</i>		
distnotice.xlsx		dstnot
<b>Generic Performance Analytics (PerfAnalytic)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
PerfAnalyticReport.xlsx		PerfData
<b>GIPS Composite Report (GIPS)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_GIPS_CompositeReport.txt		Det
rs_GIPS_CompositeReportcon.txt		Con
<i>YSR Template Names</i>		
GipsCompositereportCon.xlsx		Con
GipsCompositeReport.xlsx		Det
<b>ILPA Capital Call (ILPACap)</b>		
<i>Top-Level Select Script Files</i>		
SS_ILPA_capitalcall_header.txt		
<i>YSR Script Files</i>		
SS_ILPA_capitalcall_detail.txt		Report
<i>YSR Template Names</i>		
SS_ILPA_Capitalcall.xlsx		Report
<b>ILPA Fund Report (FundRpt)</b>		

*Top-Level Select Script Files*

SS\_YSR\_ILPA\_Script.txt

*YSR Script Files*

SS\_YSR\_ILPA\_Script.txt

ExSumm

SS\_YSR\_ILPA\_Script.txt

ILPA

*YSR Template Names*

YSR\_ILPA\_CapitalAccountStatement.xlsx

CAS

YSR\_ILPA\_Page\_21.xlsx

ExSumm

YSR\_ILPA\_Page\_BalanceSheet.xlsx

ILPA

**Investor Contact Report (investorcontact)***Top-Level Select Script Files**YSR Script Files*

rs\_investor\_contact.txt

investor

*YSR Template Names*

YSR\_InvestorContact.xlsx

investor

**NCREIF Composite Report (NCREIF)***Top-Level Select Script Files**YSR Script Files*

rs\_CompositeReport.txt

ABC

rs\_CompositeReportcon.txt

Con

*YSR Template Names*

Compositereport.xlsx

ABC

Compositereportcon.xlsx

Con

**PR Periodic Calculation Detail (Perfcaldetail)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

Perfcaldetail.xlsx

PR

**Pref\_Deal\_Listing (PR\_DList)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

DealListing.xlsx

PRDL

**Pref\_Equity Multiple (PR\_EqMul)***Top-Level Select Script Files*

*YSR Script Files**YSR Template Names*

IRRHurdleAndEquityReport.xlsx

PRDL

**Pref\_Hypothetical\_Adjustment (PR\_HypCa)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

HypoAdjustmentCalc.xlsx

PRDL

**Pref\_IRR (PR\_IRR)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

IRRHurdleAndEquityReport.xlsx

PRDL

**Pref\_PerfCalculationSummary\_PeriodAsColumn (PR\_PCSPC)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

PerfCalculationSummary\_PeriodAsColumn.xlsx

PRDL

**Pref\_Periodic\_Calculation\_Summary (PR\_PCSum)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

PerfCalculationSummary.xlsx

PRDL

**Pref\_PeriodicRuleSetup (PR\_PRS)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

PeriodicRuleSetup.xlsx

PRDL

**Pref\_SpecialDistribution (PR\_SDist)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

SpecialDistribution.xlsx

PRDL

**Pref\_SpecialRuleSetup (PR\_SDRS)**

***Top-Level Select Script Files******YSR Script Files******YSR Template Names***

SpecialRuleSetup.xlsx

PRDL

**Preferred Returns (prefret)*****Top-Level Select Script Files***

rs\_header\_deal.txt

***YSR Script Files***

rs\_detail\_deal.txt

prefret

***YSR Template Names***

prefret.xlsx

prefret

**Standard Capital Call Notice (stdcapcall)*****Top-Level Select Script Files***

rs\_stdcapcall\_header.txt

***YSR Script Files***

rs\_stdcapcall\_detail.txt

scapcall

***YSR Template Names***

stdcapcall.xlsx

scapcall

**Standard Capital Call Notice (scapcall)*****Top-Level Select Script Files******YSR Script Files***

rs\_scapcall.txt

scapcall

***YSR Template Names***

scapcall.xlsx

scapcall

**Transaction Detail Report Investment View (tran\_invest)*****Top-Level Select Script Files******YSR Script Files******YSR Template Names***

TranDetInvest.xlsx

traninve

**Transaction Detail Report Investor View (Tran\_det)*****Top-Level Select Script Files******YSR Script Files******YSR Template Names***

Tran\_Detail.xlsx

tran\_det

**Transaction Type Percentage (tranper)**

*Top-Level Select Script Files*

rs\_tranper\_header.txt

*YSR Script Files*

rs\_tranperc\_detail.txt

tranper

*YSR Template Names*

rs\_tranper.xlsx

tranper

**Transaction Types Report (trantype)***Top-Level Select Script Files*

rs\_header\_trantyp.txt

*YSR Script Files*

rs\_detail\_trantyp.txt

trantyp

*YSR Template Names*

rs\_trantypes.xlsx

trantyp

**KUBE****1001\_YSRKUBEINV Report for Kube Invoice (1001\_YSRKUBEINV)***Top-Level Select Script Files*

Kube\_Invoice\_query.txt

*YSR Script Files*

Kube\_Invoice\_query.txt

inv

*YSR Template Names*

Kube\_Invoice.docx

inv

**Kube Charge Report (1003\_KUBECHARGE)***Top-Level Select Script Files*

Kube\_Charge\_Invoice\_query.txt

*YSR Script Files*

Kube\_Charge\_Invoice\_query.txt

charge

*YSR Template Names*

Kube\_Invoice.docx

charge

**Kube Company Ledger Report (1002\_KUBECLEDGER)***Top-Level Select Script Files*

YSR\_kube\_company\_ledger\_sql.txt

*YSR Script Files*

YSR\_kube\_company\_ledger\_sql.txt

cledger

*YSR Template Names*

YSR_kube_company_ledger.xlsx	ledger
<b>Kube Proposal Report (1004_KUBEPROPOS)</b>	
<i>Top-Level Select Script Files</i>	
YSR_kube_Proposal_sql.txt	
<i>YSR Script Files</i>	
YSR_kube_Proposal_sql.txt	proposal
<i>YSR Template Names</i>	
YSR_kube_Proposal.xlsx	proposal

## Legal

### Custom Legal Global Status Change report (YSR\_LegalGSU)

<i>Top-Level Select Script Files</i>	
rs_ft_LegalCustomNoticeFilter_YSR.txt	
<i>YSR Script Files</i>	
rs_ft_LegalCustomNoticeFilter_YSR.txt	Legal
<i>YSR Template Names</i>	
select case when '#rptname#' = " then 'YSR_LegalGSU.docx' else '#rptname#' end	Legal

### Legal Collections (LegalCollections)

<i>Top-Level Select Script Files</i>	
rs_ft_LegalCustomNoticeFilter_YSR.txt	
<i>YSR Script Files</i>	
YSR_Legal_Collections.txt	LglColl
<i>YSR Template Names</i>	
YSR_Legal_Collections.xlsx	LglColl

### Legal Custom Notice in YSR (LegalCustmNotice)

<i>Top-Level Select Script Files</i>	
rs_Legal_Custom_Note_YSR.txt	
<i>YSR Script Files</i>	
rs_Legal_Custom_Note_YSR.txt	LglNotic
<i>YSR Template Names</i>	
Select Case '#smodule#' When 'Affordable Housing' then 'YSR_LegalNoticeNewVersion_AFF.xlsx'	LglNotic
when 'Public Housing' then 'YSR_LegalNoticeNewVersion_PHA.xlsx' Else	
'YSR_LegalNoticeNewVersion_RES.xlsx' end	

### Legal Notice in YSR (LegalNotice)

<i>Top-Level Select Script Files</i>	
rs_Legal_Note_YSR.txt	

*YSR Script Files*

rs_Legal_Note_YSR.txt	LglNotic
-----------------------	----------

*YSR Template Names*

Select Case '#smodule#' When 'Affordable Housing' then 'YSR_LegalNoticeNewVersion_AFF.xlsx' when 'Public Housing' then 'YSR_LegalNoticeNewVersion_PHA.xlsx' Else 'YSR_LegalNoticeNewVersion_RES.xlsx' end	LglNotic
---	----------

**Start Legal (StartLegalNotice)***Top-Level Select Script Files*

rs_Start_Legal_Note_YSR.txt
-----------------------------

*YSR Script Files*

rs_Start_Legal_Note_YSR.txt	LglNotic
-----------------------------	----------

*YSR Template Names*

Select Case '#smodule#' When 'Affordable Housing' then 'YSR_StartLegalNotice_AFF.xlsx' when 'Public Housing' then 'YSR_StartLegalNotice_PHA.xlsx' Else 'YSR_StartLegalNotice_RES.xlsx' end	LglNotic
---	----------

**Military****ArmyDash (ArmyDash)***Top-Level Select Script Files**YSR Script Files*

rs_mil_ArmyDash.txt	Army
---------------------	------

*YSR Template Names*

rx_mil_ArmyDash.xlsx	Army
----------------------	------

**NavyFlash (NavyFlash)***Top-Level Select Script Files**YSR Script Files*

rs_mil_NavyFlash.txt	Navy
----------------------	------

*YSR Template Names*

rx_mil_NavyFlash.xlsx	Navy
-----------------------	------

**New York****Resident Rent Bill (NYRentBill)***Top-Level Select Script Files**YSR Script Files*

RS_RentBill_YSR.txt	RentBill
---------------------	----------

*YSR Template Names*

RentBill\_YSR.xlsx

RentBill

## Oil and Gas

### OG Accounts Payable Detailed Aging (APAGDET)

*Top-Level Select Script Files*

rs\_OG\_APAGDET.txt

*YSR Script Files*

rs\_OG\_APAGDET.txt

APAG

*YSR Template Names*

ydoc\_OG\_APAGDet.xlsx

APAG

### OG Accounts Payable Invoice Tracking (APINVTR)

*Top-Level Select Script Files*

rs\_OG\_APINVTR.txt

*YSR Script Files*

rs\_OG\_APINVTR.txt

APINVTR

*YSR Template Names*

ydoc\_OG\_APINVTR.xlsx

APINVTR

### OG Accounts Payable Summary Aging (Invoices Paid and Unpaid)

*Top-Level Select Script Files*

rs\_OG\_APAGSUM.txt

*YSR Script Files*

rs\_OG\_APAGSUM.txt

apsum

*YSR Template Names*

ydoc\_OG\_APAGSUM.xlsx

apsum

### OG Annual Net Income Comparison Report (ANNICOMP)

*Top-Level Select Script Files*

rs\_OG\_ANNICOMP.txt

*YSR Script Files*

rs\_OG\_ANNICOMP.txt

anicomp

*YSR Template Names*

ydoc\_OG\_ANNICOMP.xlsx

anicomp

### OG Annual Review Recommendations Report (ANNREVRC)

*Top-Level Select Script Files*

rs\_OG\_ANNREVRC.txt

*YSR Script Files*

rs\_OG\_ANNREVRC.txt  
ANNREVRC

*YSR Template Names*  
ydoc\_OG\_ANNREVRC.xlsx  
ANNREVRC

#### **OG Annual Review Report (ANNREPRO)**

<i>Top-Level Select Script Files</i>	
rs_OG_ANNREPRO.txt	
<i>YSR Script Files</i>	
rs_OG_ANNREPRO.txt	
ANNREPRO	
rs_OG_ANNREPRO.txt	Summary
rs_OG_ANNREPRO_FeeCalcM.txt	FeeCalc
rs_OG_ANNREPRO_PropSummary.txt	Prod
rs_OG_ANNREPRO_PropSummary.txt	PropCnt
rs_OG_ANNREPRO_PropSummary.txt	PropCnt
PropSumm	
<i>YSR Template Names</i>	
ydoc_OG_ANNREPRO.xlsx	
ANNREPRO	
ydoc_OG_ANNREPRO_FeeCalcM.xlsx	FeeCalc
ydoc_OG_ANNREPRO_Production.xlsx	Prod
ydoc_OG_ANNREPRO_PropCount.xlsx	PropCnt
ydoc_OG_ANNREPRO_PropSummary.xlsx	PropCnt
PropSumm	
ydoc_OG_ANNREPRO_summary.xlsx	Summary

#### **OG Annual Review Report Trust List (ANNREVTL)**

<i>Top-Level Select Script Files</i>	
rs_YSR_OG_ANNREVTL.txt	
<i>YSR Script Files</i>	
rs_YSR_OG_ANNREVTL.txt	
ANNREVTL	
<i>YSR Template Names</i>	
rs_YSR_OG_ANNREVTL.xlsx	
ANNREVTL	

#### **OG AP Cross Reference By Trust (APXREF)**

<i>Top-Level Select Script Files</i>	
rs_OG_ApXRef.txt	
<i>YSR Script Files</i>	

rs_OG_ApXRef.txt	APXENT
rs_OG_ApXRef.txt	APXREF

*YSR Template Names*

ydoc_OG_ApXEntity.docx	APXENT
ydoc_OG_ApXRef.docx	APXREF

**OG AP Cross Reference By Trust Excel Output (APXEXCEL)***Top-Level Select Script Files*

rs\_OG\_ApXRefExcel.txt

*YSR Script Files*

rs\_OG\_ApXRefExcel.txt

APXEXCEL

*YSR Template Names*

ydoc\_OG\_ApXRef.xlsx

APXEXCEL

**OG AP Line Count Statistics (APLINE)***Top-Level Select Script Files**YSR Script Files*

rs\_OG\_APLINE.txt

APLINE

*YSR Template Names*

ydoc\_OG\_APLINE.xlsx

APLINE

**OG AP Proof List (APPROOF)***Top-Level Select Script Files*

rs\_OG\_APProof.txt

*YSR Script Files*

rs\_OG\_APProof.txt

APBatch

rs\_OG\_APProof.txt

APDet

rs\_OG\_APProof.txt

APSumm

*YSR Template Names*

ydoc\_OG\_APBatch.docx

APBatch

ydoc\_OG\_APDetail.docx

APDet

ydoc\_OG\_APSummary.docx

APSumm

**OG Cash Advance Tracking Detail (CASHADV)***Top-Level Select Script Files*

rs\_OG\_CASHADV.txt

*YSR Script Files*

rs\_OG\_CASHADV.txt

CashAdHe

rs_OG_CASHADVD.txt	CashAdv
<i>YSR Template Names</i>	
ydoc_OG_CASHADVDHEADER.docx	
CashAdHe	
ydoc_OG_CASHADVD.docx	CashAdv
<b>OG Cash Advance Tracking Summary (CASHADVS)</b>	
<i>Top-Level Select Script Files</i>	
rs_OG_CASHADVS.txt	
<i>YSR Script Files</i>	
rs_OG_CASHADVS.txt	CashAdv
<i>YSR Template Names</i>	
ydoc_OG_CASHADVS.xlsx	CashAdv
<b>OG Check and Invoice Activity (CKINACT)</b>	
<i>Top-Level Select Script Files</i>	
rs_OG_CKINACT.txt	
<i>YSR Script Files</i>	
rs_OG_CKINACT.txt	ckindet
rs_OG_CKINACT.txt	ckinsum
<i>YSR Template Names</i>	
ydoc_OG_CKINACT_details.xlsx	ckindet
ydoc_OG_CKINACT_Summary.xlsx	ckinsum
<b>OG Check and Invoice Activity - Summary by Purchaser/Vendor</b>	
<i>Top-Level Select Script Files</i>	
rs_OG_CKINVACT.txt	
<i>YSR Script Files</i>	
rs_OG_CKINVACT.txt	INVACT
<i>YSR Template Names</i>	
ydoc_OG_CKINVACT.xlsx	INVACT
<b>OG Closed Accounts (CLOACCT)</b>	
<i>Top-Level Select Script Files</i>	
rs_OG_CLOACCT.txt	
<i>YSR Script Files</i>	
rs_OG_CLOACCT.txt	CLOACCT
<i>YSR Template Names</i>	
ydoc_OG_CLOACCT.xlsx	CLOACCT
<b>OG Daily Cash Detail (DCASH)</b>	

*Top-Level Select Script Files*

rs\_OG\_DailyCash.txt

*YSR Script Files*

rs\_OG\_DailyCash.txt

DCASH

*YSR Template Names*

ydoc\_OG\_DailyCash.xlsx

DCASH

**OG Daily Cash Summary (DCASHS)***Top-Level Select Script Files*

rs\_OG\_DailyCashSum.txt

*YSR Script Files*

rs\_OG\_DailyCashSum.txt

DCASHS

*YSR Template Names*

ydoc\_OG\_DailyCashSum.xlsx

DCASHS

**OG Depletion Limitation (DEPLIM)***Top-Level Select Script Files*

rs\_OG\_DEPLIM.txt

*YSR Script Files*

rs\_OG\_DEPLIM.txt

DEPLIM

rs\_OG\_DEPLIM.txt

Summary

*YSR Template Names*

ydoc\_OG\_DEPLIM.xlsx

DEPLIM

ydoc\_OG\_DEPLIM\_summary.xlsx

Summary

**OG Division of Interest List (DOILIST)***Top-Level Select Script Files*

rs\_OG\_DOIList.txt

*YSR Script Files*

rs\_OG\_DOIList.txt

doi

*YSR Template Names*

ydoc\_OG\_DOIList.xlsx

doi

**OG Division Orders Aging Non Executed No Payment Received***Top-Level Select Script Files*

rs\_OG\_DOAging.txt

*YSR Script Files*

rs\_OG\_DOAging.txt

DOAGING

*YSR Template Names*

ydoc\_OG\_DOAging.xlsx  
DOAGING

#### **OG Division Orders by Entity (DOTRUST)**

*Top-Level Select Script Files*

rs\_OG\_DOTrust.txt

*YSR Script Files*

rs\_OG\_DOTrust.txt

DOTRUST

DOTSORT

*YSR Template Names*

yDoc\_OG\_DoTrustDet.docx

DOTRUST

yDoc\_OG\_DoTrustHdr.docx

DOTSORT

#### **OG Division Orders by Entity - Listing (DOTREXCL)**

*Top-Level Select Script Files*

rs\_OG\_DOTrExcl.txt

*YSR Script Files*

rs\_OG\_DOTrExcl.txt

DOTINFO

*YSR Template Names*

yDoc\_OG\_DoTrExcl.xlsx

DOTINFO

#### **OG DOs Letter Generated (DOLETGEN)**

*Top-Level Select Script Files*

rs\_OG\_DoLetGen.txt

*YSR Script Files*

rs\_OG\_DoLetGen.txt

DOLETGEN

*YSR Template Names*

ydoc\_OG\_DoLetGen.docx

DOLETGEN

#### **OG DOs Letters Generated by Purchaser (DOLETPUR)**

*Top-Level Select Script Files*

rs\_OG\_DoLetGenPurchaser.txt

*YSR Script Files*

rs\_OG\_DoLetGenPurchaser.txt

DOLETPUR

*YSR Template Names*

ydoc\_OG\_DoLetGenPurchaser.xlsx

DOLETPUR

**OG Environmental Inspection Report by Trust With Excel Output***Top-Level Select Script Files*

rs\_OG\_ENVINSP2.txt

*YSR Script Files*

rs\_OG\_ENVINSP2.txt

EnvInsp

*YSR Template Names*

ydoc\_OG\_ENVINSP\_Excel2.xlsx

EnvInsp

**OG Fee Calculation Factor Report (FEECALC)***Top-Level Select Script Files*

rs\_OG\_FeeCalc.txt

*YSR Script Files*

rs\_OG\_FeeCalc.txt

FeeCalcR

*YSR Template Names*

ydoc\_OG\_FeeCalc.xlsx

FeeCalcR

**OG Fee Calculation Report By Month (FEECALCM)***Top-Level Select Script Files*

rs\_OG\_FeeCalcM.txt

*YSR Script Files*

rs\_OG\_FeeCalcM.txt

FeeCalc

*YSR Template Names*

ydoc\_OG\_FeeCalcM.xlsx

FeeCalc

**OG Fee detail breakout by product, revenue and fee type***Top-Level Select Script Files*

rs\_OG\_FeeDet.txt

*YSR Script Files*

rs\_OG\_FeeDet.txt

FeeDeSum

rs\_OG\_FeeDet.txt

FeeDet

*YSR Template Names*

ydoc\_OG\_FeeDetSummary.xlsx

FeeDeSum

ydoc\_OG\_FeeDet.xlsx

FeeDet

**OG Fee Summary Report by Property Manager (FEESUMPM)***Top-Level Select Script Files*

rs\_OG\_FeeSumPM.txt

*YSR Script Files*

rs\_OG\_FeeSumPM.txt

Feesumpm

*YSR Template Names*

ydoc\_OG\_FeeSumPM.xlsx

Feesumpm

#### **OG Fee Variance Detail (FEEVARDE)**

*Top-Level Select Script Files*

rs\_OG\_FEEVARDE.txt

*YSR Script Files*

rs\_OG\_FEEVARDE.txt

FeeVarDe

rs\_OG\_FEEVARDE.txt

FeeVarSu

*YSR Template Names*

ydoc\_OG\_FEEVARDE.xlsx

FeeVarDe

ydoc\_OG\_FEEVARSU.xlsx

FeeVarSu

#### **OG General Liability Insurance Schedule (GENLISCH)**

*Top-Level Select Script Files*

rs\_YSR\_OG\_GenLISchedule.txt

*YSR Script Files*

rs\_YSR\_OG\_GenLISchedule.txt

GENINS

*YSR Template Names*

rs\_YSR\_OG\_GenLISchedule\_GenLiabInsurance.xlsx

GENINS

#### **OG GL Accounts and Categories (ACCTCAT)**

*Top-Level Select Script Files*

rs\_OG\_ACCTCAT.txt

*YSR Script Files*

rs\_OG\_ACCTCAT.txt

ACCTCAT

*YSR Template Names*

ydoc\_OG\_ACCTCAT.xlsx

ACCTCAT

#### **OG GL Activity Report (TRBAL)**

*Top-Level Select Script Files*

rs\_OG\_TrialBalance.txt

*YSR Script Files*

rs\_OG\_TrialBalance.txt

Triab

*YSR Template Names*

ydoc\_OG\_TrialBalance.xlsx

Triab

#### **OG Lease and Proposal by Entity (LEALPREN)**

*Top-Level Select Script Files*

rs\_OG\_LeaLprEntity.txt

*YSR Script Files*

rs\_OG\_LeaLprEntity.txt

LEALPREN

*YSR Template Names*

ydoc\_OG\_LeaLprEntity.xlsx

LEALPREN

**OG Lease and Proposal Extract (LEALPREX)***Top-Level Select Script Files*

rs\_OG\_LeaLprExtract.txt

*YSR Script Files*

rs\_OG\_LeaLprExtract.txt

LEALPREX

*YSR Template Names*

ydoc\_OG\_LeaLprExtract.xlsx

LEALPREX

**OG Lease Report by Entity (LEAENTT)***Top-Level Select Script Files*

rs\_OG\_LeaEntity.txt

*YSR Script Files*

rs\_OG\_LeaEntity.txt

LeaEntt

*YSR Template Names*

ydoc\_OG\_LeaEntity.xlsx

LeaEntt

**OG Market Value Change (OGMARKETVAL)***Top-Level Select Script Files*

rs\_OG\_MarketValueChange.txt

*YSR Script Files*

rs\_OG\_MarketValueChange.txt

marketv

*YSR Template Names*

ydoc\_OG\_MarketValueChange.xlsx

marketv

**OG Market Value History (MVHIST)***Top-Level Select Script Files*

rs\_OG\_MVHIST.txt

*YSR Script Files*

rs\_OG\_MVHIST.txt

MVHIST

*YSR Template Names*

ydoc_OG_MVHIST.xlsx	MVHIST
<b>OG Mineral Accounting Report - Cash Basis (MINACCT)</b>	
<i>Top-Level Select Script Files</i>	
rs_YSR_OG_MINACCT.txt	
<i>YSR Script Files</i>	
rs_YSR_OG_MINACCT.txt	actdet
rs_YSR_OG_MINACCT.txt	actgl
rs_YSR_OG_MINACCT.txt	actsum
<i>YSR Template Names</i>	
rs_YSR_OG_MINACCT_Detail.xlsx	actdet
rs_YSR_OG_MINACCT_byGl.xlsx	actgl
rs_YSR_OG_MINACCT_Summary for all asset.xlsx	actsum
<b>OG Mineral Asset (MINASSET)</b>	
<i>Top-Level Select Script Files</i>	
rs_YSR_OG_MinAssetDet.txt	
<i>YSR Script Files</i>	
rs_YSR_OG_MinAssetDet.txt	
MINASENT	
rs_YSR_OG_MinAssetDet.txt	
MINASSET	
rs_YSR_OG_MinAssetDet.txt	
MINASSUM	
<i>YSR Template Names</i>	
rs_YSR_OG_MinAssetEntity.docx	
MINASENT	
rs_YSR_OG_MinAsset.docx	
MINASSET	
rs_YSR_OG_MinAssetSummary.docx	
MINASSUM	
<b>OG Mineral Asset Details (MINASDET)</b>	
<i>Top-Level Select Script Files</i>	
rs_YSR_OG_MinAssetDet.txt	
<i>YSR Script Files</i>	
rs_YSR_OG_MinAssetDet.txt	
MINASDET	
rs_YSR_OG_MinAssetDet.txt	
MINASENT	
rs_YSR_OG_MinAssetDet.txt	
MINASSUM	

*YSR Template Names*

rs\_YSR\_OG\_MinAssetDet.docx

MINASDET

rs\_YSR\_OG\_MinAssetEntity.docx

MINASENT

rs\_YSR\_OG\_MinAssetSummary.docx

MINASSUM

**OG Mineral Principal Report (MINPRIN)***Top-Level Select Script Files*

rs\_OG\_MinPrin.txt

*YSR Script Files*

rs\_OG\_MinPrin.txt

minpdet

rs\_OG\_MinPrin.txt

minpri

*YSR Template Names*

ydoc\_OG\_MINPRIN\_Detail.xlsx

minpdet

ydoc\_OG\_MINPRIN\_Summary.xlsx

minpri

**OG Mineral Transaction Report (MINTRA)***Top-Level Select Script Files*

rs\_OG\_Mintra.txt

*YSR Script Files*

rs\_OG\_Mintra.txt

minsumma

rs\_OG\_Mintra.txt

mintragl

rs\_OG\_Mintra.txt

mintrdet

*YSR Template Names*

ydoc\_OG\_Mintra\_Summary for all asset.xlsx

minsumma

ydoc\_OG\_Mintra\_byGl.xlsx

mintragl

ydoc\_OG\_Mintra\_Detail\_AllAsset.xlsx

mintrdet

**OG Mineral Transaction Report - Excel (MINTRAEX)***Top-Level Select Script Files*

rs\_OG\_MintraEx.txt

*YSR Script Files*

rs\_OG\_MintraEx.txt

MINTRAEX

*YSR Template Names*

ydoc\_OG\_MintraEx\_SummaryandDetail.xlsx

MINTRAEX

**OG Mineral Transaction Summary (MINTRASU)***Top-Level Select Script Files*

rs\_YSR\_OG\_MintraSummary.txt

*YSR Script Files*

rs\_YSR\_OG\_MintraSummary.txt

minsumma

rs\_YSR\_OG\_MintraSummary.txt

mintragl

*YSR Template Names*

rs\_YSR\_OG\_MintraSum\_Summary for all asset.xlsx

minsumma

rs\_YSR\_OG\_MintraSum\_byGl.xlsx

mintragl

**OG MJE Proofing (MJEPROOF)***Top-Level Select Script Files*

rs\_OG\_MJEProof.txt

*YSR Script Files*

rs\_OG\_MJEProof.txt

MJEBatch

rs\_OG\_MJEProof.txt

MJEBSumm

rs\_OG\_MJEProof.txt

MJEDet

rs\_OG\_MJEProof.txt

MJEESumm

*YSR Template Names*

ydoc\_OG\_MJEBatch.docx

MJEBatch

ydoc\_OG\_MJEBatchSumm.docx

MJEBSumm

ydoc\_OG\_MJEDetail.docx

MJEDet

ydoc\_OG\_MJEEEntitySumm.docx

MJEESumm

**OG Monthly History Dtl by Field - Cash (MHDETFI)***Top-Level Select Script Files*

rs\_OG\_MHDET\_Field.txt

*YSR Script Files*

rs\_OG\_MHDET\_Field.txt

MHDFI

rs\_OG\_MHDET\_Field.txt

Summ

*YSR Template Names*

ydoc\_OG\_MthlyHisFld\_CA.xlsx

MHDFI

ydoc\_OG\_MthlyHisFld\_CASumm.xlsx

Summ

**OG Monthly History Dtl by Grouping Code - Cash (MHDETAQ)***Top-Level Select Script Files*

rs\_OG\_MHDET\_ACQ.txt

*YSR Script Files*

rs\_OG\_MHDET\_ACQ.txt

MHACQ

rs\_OG\_MHDET\_ACQ.txt

MHACQSUM

*YSR Template Names*

ydoc\_OG\_MthlyHisACQ\_Cash.xlsx

MHACQ

ydoc\_OG\_MthlyHisACQ\_Summ.xlsx

MHACQSUM

**OG Oil and Gas History - Cash Basis (OGHISCAS)***Top-Level Select Script Files*

rs\_OG\_CashBasisHistory.txt

*YSR Script Files*

rs\_OG\_CashBasisHistory.txt

CshHist

*YSR Template Names*

ydoc\_OG\_CashBasisHistory.xlsx

CshHist

**OG Payout Detail Report (PAYOUTDE)***Top-Level Select Script Files*

rs\_OG\_PayOutDet.txt

*YSR Script Files*

rs\_OG\_PayOutDet.txt

PAYOUTDE

*YSR Template Names*

ydoc\_OG\_PayOutDet.xlsx

PAYOUTDE

**OG Payout Summary Report (PAYOUTSM)***Top-Level Select Script Files*

rs\_OG\_PayOutSum.txt

*YSR Script Files*

rs\_OG\_PayOutSum.txt

PayOutSm

*YSR Template Names*

ydoc\_OG\_PayOutSum.xlsx

PayOutSm

**OG Pending Disbursements (CHKPROS)**

*Top-Level Select Script Files*

rs\_YSR\_OG\_CHKPROS.txt

*YSR Script Files*

rs\_YSR\_OG\_CHKPROS.txt

chkdet

*YSR Template Names*

rs\_YSR\_OG\_CHKPROS\_PendingDisbursements.xlsx

chkdet

**OG Production Revenue Tax Report (CUSTPRT)***Top-Level Select Script Files*

rs\_OG\_CustPrt.txt

*YSR Script Files*

rs\_OG\_CustPrt.txt

CPSTATE

rs\_OG\_CustPrt.txt

CPSTOT

rs\_OG\_CustPrt.txt

CPTTOT

rs\_OG\_CustPrt.txt

CUSTPRT

*YSR Template Names*

ydoc\_OG\_CPState.docx

CPSTATE

ydoc\_OG\_CPSTot.docx

CPSTOT

ydoc\_OG\_CPTTot.docx

CPTTOT

ydoc\_OG\_CPEntity.docx

CUSTPRT

**OG Property File Label (PROPLABL)***Top-Level Select Script Files*

rs\_OG\_PROPLABL.txt

*YSR Script Files*

rs\_OG\_PROPLABL.txt

PROPLABL

*YSR Template Names*

ydoc\_OG\_PROPLABL.docx

PROPLABL

**OG Property Summary Statement (PROPSUMM)***Top-Level Select Script Files*

rs\_OG\_PropSDtl.txt

*YSR Script Files*

rs\_OG\_PropSDtl.txt

entsum

rs\_OG\_PropSDtl.txt

Prptrdet

*YSR Template Names*

ydoc\_OG\_PropSDtl\_entity.xlsx

entsum

ydoc\_OG\_PropSumm.xlsx

Prptrdet

**OG Property Summary Statement With Detail (PROPSDTL)**

*Top-Level Select Script Files*

rs\_OG\_PropSDtl.txt

*YSR Script Files*

rs\_OG\_PropSDtl.txt

rs\_OG\_PropSDtl.txt

entsum

propsum

*YSR Template Names*

ydoc\_OG\_PropSDtl\_entity.xlsx

entsum

ydoc\_OG\_PropSDtl.xlsx

propsum

**OG Proposal Report by Entity (LPRENTT)***Top-Level Select Script Files*

rs\_OG\_LprEntity.txt

*YSR Script Files*

rs\_OG\_LprEntity.txt

LprCnt

rs\_OG\_LprEntity.txt

LprEntt

*YSR Template Names*

ydoc\_OG\_LprEntity\_Cnt.xlsx

LprCnt

ydoc\_OG\_LprEntity.xlsx

LprEntt

**OG Purchaser Cross Reference by Entity (PURCXPDF)***Top-Level Select Script Files*

rs\_OG\_Purcxref.txt

*YSR Script Files*

rs\_OG\_Purcxref.txt

PURCXENT

rs\_OG\_Purcxref.txt

PURCXWEL

*YSR Template Names*

ydoc\_OG\_PXEntity.docx

PURCXENT

ydoc\_OG\_PXWell.docx

PURCXWEL

**OG Revenue Cash Receipts Batch Validation (RCRPROOF)***Top-Level Select Script Files*

rs\_OG\_RCRProof.txt

*YSR Script Files*

rs\_OG\_RCRProof.txt

RCRPROOF

*YSR Template Names*

ydoc\_OG\_RCRProof.xlsx

RCRPROOF

**OG Revenue Line Item Detail (REVLIDET)**

*Top-Level Select Script Files*

rs\_OG\_REVLINEDET.txt

*YSR Script Files*

rs\_OG\_REVLINEDET.txt

REVDET

*YSR Template Names*

ydoc\_OG\_REVLINEDET.xlsx

REVDET

**OG Revenue Line Item Summary (REVLISUM)**

*Top-Level Select Script Files*

rs\_OG\_REVLINESUM.txt

*YSR Script Files*

rs\_OG\_REVLINESUM.txt

REVSUM

*YSR Template Names*

ydoc\_OG\_REVLINESummary.xlsx

REVSUM

**OG RVS Proofing (RVSPROOF)**

*Top-Level Select Script Files*

rs\_OG\_RVSPProof.txt

*YSR Script Files*

rs\_OG\_RVSPProof.txt

RVSBatch

rs\_OG\_RVSPProof.txt

RVSDet

rs\_OG\_RVSPProof.txt

RVSPSumm

*YSR Template Names*

ydoc\_OG\_RVSBatch.docx

RVSBatch

ydoc\_OG\_RVSDetail.docx

RVSDet

ydoc\_OG\_RVSPurcSumm.docx

RVSPSumm

**OG State Asset Listing Report (STASLIST)**

*Top-Level Select Script Files*

rs\_OG\_StateAssetList.txt

*YSR Script Files*

rs\_OG\_StateAssetList.txt

STATLIST

*YSR Template Names*

ydoc\_OG\_StateAssetList.xlsx

STATLIST

**OG Statement Recipient Report (STMTREC)**

*Top-Level Select Script Files*

rs\_OG\_STMTREC.txt

*YSR Script Files*

rs\_OG\_STMTREC.txt

STMTREC

*YSR Template Names*

ydoc\_OG\_STMTREC.xlsx

STMTREC

### **OG Summary of Income & Expenses with Principal (SUMIEPRN)**

*Top-Level Select Script Files*

rs\_OG\_SumIncExPrin.txt

*YSR Script Files*

rs\_OG\_SumIncExPrin.txt

IEPSUMM

rs\_OG\_SumIncExPrin.txt

SUMIEPRN

*YSR Template Names*

ydoc\_OG\_SumIncExPrin.xlsx

IEPSUMM

ydoc\_OG\_SumIncExPrin\_Details.xlsx

SUMIEPRN

### **OG Summary of Income and Expenses (SUMINCEX)**

*Top-Level Select Script Files*

rs\_YSR\_OG\_SumIncExDpl.txt

*YSR Script Files*

rs\_YSR\_OG\_SumIncExDpl.txt

SUMIE

rs\_YSR\_OG\_SumIncExDpl.txt

SUMINCEX

*YSR Template Names*

rs\_YSR\_OG\_SumIncExDpl.xlsx

SUMIE

rs\_YSR\_OG\_SumIncExDpl\_Details.xlsx

SUMINCEX

### **OG Summary of Income and Expenses by Field (SUMINEXF)**

*Top-Level Select Script Files*

rs\_OG\_SUMINEXF.txt

*YSR Script Files*

rs\_OG\_SUMINEXF.txt

SUMMARY

*YSR Template Names*

ydoc\_OG\_SUMINEXF.xlsx

SUMMARY

**OG Summary of Income and Expenses by Grouping Code***Top-Level Select Script Files*

rs\_OG\_SINEXACQ.txt

*YSR Script Files*

rs\_OG\_SINEXACQ.txt

SINEXACQ

rs\_OG\_SINEXACQ.txt

SUMMARY

*YSR Template Names*

ydoc\_OG\_SINEXACQ\_Details.xlsx

SINEXACQ

ydoc\_OG\_SINEXACQ\_Summary.xlsx

SUMMARY

**OG Tax Report (TAXREP)***Top-Level Select Script Files*

rs\_YSR\_OG\_TaxReport.txt

*YSR Script Files*

rs\_YSR\_OG\_TaxReport.txt

ETSB

rs\_YSR\_OG\_TaxReport.txt

STSB

rs\_YSR\_OG\_TaxReport.txt

TRE

rs\_YSR\_OG\_TaxReport.txt

TW

rs\_YSR\_OG\_TaxReport.txt

TWST

*YSR Template Names*

rs\_YSR\_OG\_TaxReport\_TXEntityTot.docx

ETSB

rs\_YSR\_OG\_TaxReport\_TXStateTot.docx

STSB

rs\_YSR\_OG\_TaxReport.docx

TRE

rs\_YSR\_OG\_TaxReport\_TXWell.docx

TW

rs\_YSR\_OG\_TaxReport\_TXWellST.docx

TWST

**OG Trial Balance Spreadsheet (TRBALROL)***Top-Level Select Script Files*

rs\_OG\_TRBALROL.txt

*YSR Script Files*

rs\_OG\_TRBALROL.txt

TRBALROL

*YSR Template Names*

ydoc\_OG\_TRBALROL.xlsx

TRBALROL

**OG Unprocessed Revenue (UNPREV)***Top-Level Select Script Files*

rs\_OG\_UnprocessedRevenue.txt

*YSR Script Files*

rs\_OG\_UnprocessedRevenue.txt

UNPREV

*YSR Template Names*

ydoc\_OG\_UnprocessedRevenue.xlsx

UNPREV

**OG Unprocessed Revenue Summary (UNPSUM)**

*Top-Level Select Script Files*

rs\_og\_UnprocessedRevenueSummary.txt

*YSR Script Files*

rs\_og\_UnprocessedRevenueSummary.txt

UNPSUM

*YSR Template Names*

ydoc\_og\_UnprocessedRevenueSummary.xlsx

UNPSUM

**OG Vendor Search (VENDSRCH)**

*Top-Level Select Script Files*

rs\_OG\_VendorSearch.txt

*YSR Script Files*

rs\_OG\_VendorSearch.txt

vendor

*YSR Template Names*

ydoc\_og\_VendorSearch.xlsx

vendor

## P2P

**AP Report Card (P2PScore)**

*Top-Level Select Script Files*

*YSR Script Files*

rs\_YSR\_APRC.txt

APRC

*YSR Template Names*

APRC.xlsx

APRC

**PAYscan Operations Analytics (POAp)**

*Top-Level Select Script Files*

*YSR Script Files*

rs\_YSR\_POA.txt

POA

*YSR Template Names*

POA.xlsx

POA

**YMP and VC Potential Savings Analysis (SavingsAnalysis)**

*Top-Level Select Script Files*

*YSR Script Files*

YSR\_VcYmpSavingsAnalysis\_v1\_00.txt SavA

*YSR Template Names*

YSR\_VcYmpSavingsAnalysis.xls SavA

## PayScan

**PAYscan Operations Analytics (POAi)***Top-Level Select Script Files**YSR Script Files*

rs\_YSR\_POAi.txt POA

*YSR Template Names*

POAi.xlsx POA

**PAYscan Operations Analytics (Summary) (POA\_Summary)***Top-Level Select Script Files**YSR Script Files*

rs\_YSR\_POAp.txt POA

*YSR Template Names*

POAp.xlsx POA

## PHA

**Active Units (ActiveUnits)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_Active\_Units\_YSR.txt ActDet

*YSR Template Names*

rx\_PHA\_Active\_Units\_YSR.xlsx ActDet

**Applicant Registration Letter (Applicant\_Invite)***Top-Level Select Script Files*

rs\_PHA\_HC\_YSR\_Applicant\_Registration\_Letter.txt

*YSR Script Files*

rs\_PHA\_HC\_YSR\_Applicant\_Registration\_Letter.txt TenList

*YSR Template Names*

rx\_PHA\_HC\_YSR\_Applicant\_Registration\_Letter.docx TenList

**Appointment Letters (InitAppt)**

*Top-Level Select Script Files*

rs\_PHA\_YSR\_Appointment\_Letter.txt

*YSR Script Files*

rs\_PHA\_YSR\_Appointment\_Letter.txt

apptltr

rs\_PHA\_YSR\_Appointment\_Letter.txt

InitAppt

*YSR Template Names*

rx\_PHA\_YSR\_Appointment\_Letter.docx

apptltr

rx\_PHA\_YSR\_Appointment\_Letter\_Initial.docx

InitAppt

**Batch Inspections (BatchInspection)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_YSR\_BatchInspectionSchd.txt

InspSchd

*YSR Template Names*

SELECT CASE '#RptName#' WHEN 'Batch Inspections-Move Out' THEN

InspSchd

'rx\_PHA\_YSR\_BatchInspectionMoveOut.XLSX' WHEN 'Batch Inspections-Property' THEN

'rx\_PHA\_YSR\_BatchInspectionProperty.XLSX' WHEN 'Batch Inspections-Inspection Cycle' THEN

'rx\_PHA\_YSR\_BatchInspectionCycle.XLSX' END

**Bonus - Voucher Census Demographics (SEMAPBVD)***Top-Level Select Script Files*

rs\_PHA\_YSR\_Census\_Demographics.txt

*YSR Script Files*

rs\_PHA\_YSR\_Census\_Demographics.txt

BVCD

rs\_PHA\_YSR\_Census\_Demographics.txt

BVCDG

rs\_PHA\_YSR\_Census\_Demographics.txt

BVCDL

*YSR Template Names*

rx\_PHA\_YSR\_Census\_Demographics\_CrossTab.xlsx

BVCD

rx\_PHA\_YSR\_Census\_Demographics\_Graphical.xlsx

BVCDG

rx\_PHA\_YSR\_Census\_Demographics\_Listing.xlsx

BVCDL

**Earned Income Disallowance (EID) Status Change Letter (EIDStsCh)***Top-Level Select Script Files*

rs\_PHA\_YSR\_EID\_Status\_Change\_Letter.txt

*YSR Script Files*

rs\_PHA\_YSR\_EID\_Status\_Change\_Letter.txt

EIDStsCh

*YSR Template Names*

rx\_PHA\_YSR\_Notece\_Of\_EIDStatusChange.doc

EIDStsCh

**Family Review (FamReview)***Top-Level Select Script Files*

rs\_PHA\_YSR\_FamilyReview.txt

*YSR Script Files*

rs\_PHA\_YSR\_FamilyReview.txt

FamRev

*YSR Template Names*

rx\_PHA\_YSR\_FamilyReview.docx

FamRev

#### **FSS Escrow Statement (FSSEscrowStat)**

*Top-Level Select Script Files*

rs\_PHA\_YSR\_FSS\_Statement.txt

*YSR Script Files*

rs\_PHA\_YSR\_FSS\_Statement.txt

FSSEscro

*YSR Template Names*

rx\_PHA\_YSR\_FSS\_Statement.docx

FSSEscro

#### **FSS Participants Escrow Report (FSSEscrw)**

*Top-Level Select Script Files*

rs\_PHA\_YSR\_FSS\_Participants\_Escrow.txt

*YSR Script Files*

rs\_PHA\_YSR\_FSS\_Participants\_Escrow.txt

FSSE

*YSR Template Names*

SELECT CASE '#sRptType#' WHEN 'Detail' THEN

FSSE

'rx\_PHA\_YSR\_FSS\_Participants\_Escrow\_Detail.xlsx' ELSE

#### **General Ledger by Transaction Month (GLByMon)**

*Top-Level Select Script Files*

rs\_PHA\_YSR\_General\_Ledger\_CAT\_Month.txt

*YSR Script Files*

rs\_PHA\_YSR\_General\_Ledger\_CAT\_Month.txt

GLByMon

*YSR Template Names*

SELECT CASE '#sSummary#' WHEN 'Yes' THEN 'rx\_PHA\_YSR\_General\_Ledger\_CAT\_Month' WHEN

GLByMon

'No' THEN 'rx\_PHA\_YSR\_General\_Ledger\_CAT\_Month\_Detail' ELSE

'rx\_PHA\_YSR\_General\_Ledger\_CAT\_Month' END + '.xlsx'

#### **HCV Two Year Forecasting Tool (HUDForecasting)**

*Top-Level Select Script Files*

*YSR Script Files*

rs\_PHA\_YSR\_HUDForecasting.txt

Forecast

*YSR Template Names*

rx_PHA_YSR_HUDForecasting.xlsx	Forecast
<b>Income Listing- Single Income Type (Inclstng)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_IncomeList.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_IncomeList.txt	Inclstng
<i>YSR Template Names</i>	
SELECT CASE WHEN '#SortOrd#' in ('Tenant Code', 'Tenant Name', 'Tenant SSN') THEN 'rx_PHA_YSR_IncomeList_For_Tenants.xlsx' ELSE 'rx_PHA_YSR_IncomeList.xlsx' END	Inclstng
<b>Inspection Checklist - HUD 52580 (IChkLst)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_Voucher_Inspection_Checklist_HUD52580.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_Voucher_Inspection_Checklist_HUD52580.txt	IChkLst
<i>YSR Template Names</i>	
rx_PHA_YSR_Voucher_Inspection_Checklist_HUD52580.xlsx	IChkLst
<b>Inspection Long Form - HUD 52580A - Landlord (InspLongFormLL)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_InspectionLongForm_52580a.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_InspectionLongForm_52580a.txt	insplfll
<i>YSR Template Names</i>	
rx_PHA_YSR_InspectionLongForm_52580a.docx	insplfll
<b>Inspection Long Form - HUD 52580A - Tenant (InspLongForm)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_InspectionLongForm_52580a.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_InspectionLongForm_52580a.txt	inspform
<i>YSR Template Names</i>	
rx_PHA_YSR_InspectionLongForm_52580a.docx	inspform
<b>Interview Letter (FSSInterviewltr)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_FSS_Interview_Letter_1.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_FSS_Interview_Letter_1.txt	FSSInter
<i>YSR Template Names</i>	

rx_PHA_YSR_FSS_Interview_Letter.doc	FSSI
<b>Landlord Lease Amendment (VenLeaseAmendmnt)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_Lease_Amendment_Landlord.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_Lease_Amendment_Landlord.txt	Leaseam
<i>YSR Template Names</i>	
rx_PHA_YSR_Lease_Amendment.docx	Leaseam
<b>Landlord Overpayment Letter (LOPList)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_Landlord_Overpayment_Listing.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_Landlord_Overpayment_Listing.txt	LOPList
<i>YSR Template Names</i>	
rx_PHA_YSR_Landlord_Overpayment_Listing_Letter.docx	LOPList
<b>Landlord Recertification Letters (RecertLetOwn)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_Recert_Owner.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_Recert_Owner.txt	RecertLe
<i>YSR Template Names</i>	
rx_PHA_YSR_Recert_Owner.docx	RecertLe
<b>One Way Referral Letter (1WayRefr)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_OneWayReferralLetter.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_OneWayReferralLetter.txt	
1WayRefr	
<i>YSR Template Names</i>	
rx_PHA_YSR_One_Way_Referral_Letter.docx	
<b>Other Housing Authority Listing (OHAList)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_OtherHA_Listing.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_OtherHA_Listing.txt	OHAList

*YSR Template Names*

Select 'rx\_PHA\_YSR\_OtherHA\_Listing\_' + case when '#RptType#'='Summary' then 'Summary' else  
 'Details' end + '.xlsx'

**PBV Statement of Family Responsibility HUD 52578B (PBV52578)***Top-Level Select Script Files*

rs\_PHA\_YSR\_PBV\_StmtOfFamResponsibility\_HUD52578B.txt

*YSR Script Files*

rs\_PHA\_YSR\_PBV\_StmtOfFamResponsibility\_HUD52578B.txt

PBV52578

*YSR Template Names*

rx\_PHA\_YSR\_PBV\_StmtOfFamilyResposibility\_HUD52578B.docx

PBV52578

**PHA Automated Unit Offer Withdrawal Letter (UOfrWtdr)***Top-Level Select Script Files*

rs\_PHA\_YSR\_AutomatedUnitOffer\_Withdrawal\_Letter.txt

*YSR Script Files*

rs\_PHA\_YSR\_AutomatedUnitOffer\_Withdrawal\_Letter.txt

UOfrWtdr

*YSR Template Names*

rx\_PHA\_YSR\_AutomatedUnitOffer\_Withdrawal\_Letter.docx

UOfrWtdr

**PHA Defaults & Options and Special Programs Settings (DNO)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_PHADefNOptList.txt basic

*YSR Template Names*

rx\_PHA\_YSR\_DefNOptList.xlsx basic

**PHA Demographics (PHADemo)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_YSR\_Demographics.txt

PHADemo

*YSR Template Names*

rx\_PHA\_YSR\_Demographics.xlsx

PHADemo

**PHA Key Management Indicators (PHAKMII)***Top-Level Select Script Files*

rs\_PHA\_YSR\_Key\_Management\_Indicators.txt

*YSR Script Files*

rs\_PHA\_YSR\_Key\_Management\_Indicators.txt

KMI

*YSR Template Names*

SELECT 'rx\_PHA\_YSR\_'+ Case When '#Graphs#' ='Yes' Then 'KeyManagementIndicators' Else  
'KeyManagementIndicators1' End+'.xlsx'

KMI

**PHA Level of Behavior Modification Report (LvlBehav)**

*Top-Level Select Script Files*

rs\_PHA\_YSR\_Level\_of\_Behavior\_Modification.txt

*YSR Script Files*

rs\_PHA\_YSR\_Level\_of\_Behavior\_Modification.txt

LvlBehav

*YSR Template Names*

rx\_PHA\_YSR\_Level\_of\_Behavior\_Modification.docx

LvlBehav

**PHA MTW Work Request Violation Letter (WorkReqV)**

*Top-Level Select Script Files*

rs\_PHA\_YSR\_MTW\_WorkReqViolation\_Letter.txt

*YSR Script Files*

rs\_PHA\_YSR\_MTW\_WorkReqViolation\_Letter.txt

WorkReqV

*YSR Template Names*

rx\_PHA\_YSR\_MTW\_WorkReqViolation\_Letter.docx

WorkReqV

**PHA MTW Work Request Warning Letter (WorkReqW)**

*Top-Level Select Script Files*

rs\_PHA\_YSR\_MTW\_WorkReqWarning\_Letter.txt

*YSR Script Files*

rs\_PHA\_YSR\_MTW\_WorkReqWarning\_Letter.txt

WorkReqW

*YSR Template Names*

rx\_PHA\_YSR\_MTW\_WorkReqWarning\_Letter.docx

WorkReqW

**PHA Payment Summary (PHAPaymtSummar)**

*Top-Level Select Script Files*

rs\_PHA\_YSR\_Payment\_Summary.txt

*YSR Script Files*

rs\_PHA\_YSR\_Payment\_Summary.txt

Primary

*YSR Template Names*

rx_PHA_YSR_Payment_Summary.xlsx	Primary
<b>PHA Recert Outcome Report (RecertOutcome)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_PHA_YSR_RecertOutcome.txt	basic
<i>YSR Template Names</i>	
rx_PHA_YSR_RecertOutcome.xlsx	basic
<b>PHA Resident Characteristics Report (PHARCR)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_Resident_Characteristics.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_Resident_Characteristics.txt	RCR
<i>YSR Template Names</i>	
SELECT 'rx_PHA_YSR_Resident_'+ Case When '#Graphs#' ='Yes' Then 'Characteristics' Else 'Characteristics1' End+'.xlsx'	RCR
<b>PHA Setup Review (PHASetup)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_PHA_YSR_Analysis.txt	PropSet
<i>YSR Template Names</i>	
rx_PHA_YSR_Analysis.xlsx	PropSet
<b>PHA Waiting List Contact Remove Letter (WLRemove)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_Contact_Remove_Letter.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_Contact_Remove_Letter.txt	
WLRemove	
<i>YSR Template Names</i>	
rx_PHA_YSR_NoResponse_Removal_Letter.docx	
WLRemove	
<b>Portability Billing Statement (PortBill)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_Portability_Statement.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_Portability_Statement.txt	PortBill
rs_PHA_YSR_Portability_Statement.txt	Transact

*YSR Template Names*

rx\_PHA\_YSR\_Portability\_Statement.xlsx  
rx\_PHA\_YSR\_Portability\_Statement\_Transact.xlsx

PortBill  
Transact

**Post Subsidies After-Post Report (PHAPRHAftrPost)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_YSR\_PRH\_AfterPostingRpt.txt

PHAPRHAf

*YSR Template Names*

rx\_PHA\_YSR\_PRH\_EditList.xlsx

PHAPRHAf

**Post Subsidies Audit Report (PHAPRHAAudit)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_YSR\_PRH\_AuditRpt.txt

PHAPRHAAu

*YSR Template Names*

rx\_PHA\_YSR\_PRH\_Audit.xlsx

PHAPRHAAu

**Post Subsidies Delta Report (PRHDelta)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_YSR\_PRH\_DeltaRpt.txt

PRHDelta

*YSR Template Names*

rx\_PHA\_YSR\_PRH\_DeltaRpt.xlsx

PRHDelta

**Post Subsidies Edit List (PHAPRHedit\_1)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_YSR\_PRH\_EditList.txt

prhedit

*YSR Template Names*

rx\_PHA\_YSR\_PRH\_EditList.xlsx

prhedit

**Post Subsidies Edit List (PHAPRHedit)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_YSR\_PRH\_EditList.txt

prhedit

*YSR Template Names*

rx_PHA_YSR_PRH_EditList.xlsx	prhedit
<b>Post Subsidies Pre-Posting Report (PHAPRHPo)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_PHA_YSR_PRH_PostingRpt.txt	
PHAPRHPo	
<i>YSR Template Names</i>	
rx_PHA_YSR_PRH_EditList.xlsx	
PHAPRHPo	
<b>Post Subsidies Transaction Recap Report (PRHTranR)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_PRH_TransactionRecap.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_PRH_TransactionRecap.txt	FSS
rs_PHA_YSR_PRH_TransactionRecap.txt	HAP
rs_PHA_YSR_PRH_TransactionRecap.txt	HAPChrg
rs_PHA_YSR_PRH_TransactionRecap.txt	HTH
rs_PHA_YSR_PRH_TransactionRecap.txt	OTF
rs_PHA_YSR_PRH_TransactionRecap.txt	PAF
rs_PHA_YSR_PRH_TransactionRecap.txt	POA
rs_PHA_YSR_PRH_TransactionRecap.txt	POH
rs_PHA_YSR_PRH_TransactionRecap.txt	PRT
rs_PHA_YSR_PRH_TransactionRecap.txt	TPA
rs_PHA_YSR_PRH_TransactionRecap.txt	TRC
rs_PHA_YSR_PRH_TransactionRecap.txt	TRPort
rs_PHA_YSR_PRH_TransactionRecap.txt	TrRec
rs_PHA_YSR_PRH_TransactionRecap.txt	URP
<i>YSR Template Names</i>	
rx_PHA_YSR_PRH_TransactionRecap_FSSDet.xlsx	FSS
rx_PHA_YSR_PRH_TransactionRecap_HAPDet.xlsx	HAP
rx_PHA_YSR_PRH_TransactionRecap_HAPChrgesDet.xlsx	HAPChrg
rx_PHA_YSR_PRH_TransactionRecap_HTHDet.xlsx	HTH
rx_PHA_YSR_PRH_TransactionRecap_OTFDet.xlsx	OTF
rx_PHA_YSR_PRH_TransactionRecap_PAFDet.xlsx	PAF
rx_PHA_YSR_PRH_TransactionRecap_POADet.xlsx	POA
rx_PHA_YSR_PRH_TransactionRecap_POHDet.xlsx	POH
rx_PHA_YSR_PRH_TransactionRecap_PRTDet.xlsx	PRT
rx_PHA_YSR_PRH_TransactionRecap_TPADet.xlsx	TPA
rx_PHA_YSR_PRH_TransactionRecap_TRCDet.xlsx	TRC
rx_PHA_YSR_PRH_TransactionRecapPort.xlsx	TRPort

rx_PHA_YSR_PRH_TransactionRecap.xlsx	TrRec
rx_PHA_YSR_PRH_TransactionRecap_URPDet.xlsx	URP

**Pre Check/EFT Processing Report (PrepayReport)***Top-Level Select Script Files**YSR Script Files*

rs_PHA_YSR_PrePayReport.txt	Prepay
-----------------------------	--------

*YSR Template Names*

```
SELECT CASE WHEN '#RptType#' = 'Summary' THEN 'rx_PHA_YSR_PrePayReport_Summary.xlsx'
ELSE 'rx_PHA_YSR_PrePayReport_Detail.xlsx' END
```

**Pre Check/EFT Processing Report (PrepayReport\_1)***Top-Level Select Script Files**YSR Script Files*

rs_PHA_YSR_PrePayReport.txt	Prepay
-----------------------------	--------

*YSR Template Names*

rx_PHA_YSR_PrePayReport_Detail.xlsx	Prepay
-------------------------------------	--------

**PRH Abatement Completion Letter - Landlord (AbtmtLTR\_1)***Top-Level Select Script Files*

rs_PHA_YSR_AbatementLetter.txt
--------------------------------

*YSR Script Files*

rs_PHA_YSR_AbatementLetter.txt
--------------------------------

AbtmtLTR

*YSR Template Names*

rx_PHA_YSR_AbatementLetter.doc
--------------------------------

AbtmtLTR

**PRH Abatement Completion Letter - Landlord (AbtmtLTR)***Top-Level Select Script Files*

rs_PHA_YSR_AbatementLetter.txt
--------------------------------

*YSR Script Files*

rs_PHA_YSR_AbatementLetter.txt
--------------------------------

AbtmtLTR

*YSR Template Names*

rx_PHA_YSR_AbatementLetter.doc
--------------------------------

AbtmtLTR

**PRH Abatement Completion Letter - Tenant (AbtmtLRTen)***Top-Level Select Script Files*

rs_PHA_YSR_AbatementLetter_Tenant.txt
---------------------------------------

*YSR Script Files*

rs\_PHA\_YSR\_AbatementLetter\_Tenant.txt  
AbtmtLTR

*YSR Template Names*  
rx\_PHA\_YSR\_AbatementLetter.doc  
AbtmtLTR

#### **PRH Abatement Completion Letter - Tenant (AbtmtLRTen\_1)**

*Top-Level Select Script Files*  
rs\_PHA\_YSR\_AbatementLetter\_Tenant.txt

*YSR Script Files*  
rs\_PHA\_YSR\_AbatementLetter\_Tenant.txt  
AbtmtLTR

*YSR Template Names*  
rx\_PHA\_YSR\_AbatementLetter.doc  
AbtmtLTR

#### **PRH Scheduling and Posting Settings (PRHsett)**

*Top-Level Select Script Files*

*YSR Script Files*  
rs\_PHA\_YSR\_PRHSettings.txt

YSR

*YSR Template Names*  
rx\_PHA\_YSR\_PRHSettings.xlsx

YSR

#### **Public Housing Lease (PHLease)**

*Top-Level Select Script Files*  
rs\_PHA\_YSR\_PublicHousing\_Lease.txt

*YSR Script Files*  
rs\_PHA\_YSR\_PublicHousing\_Lease.txt

PHLease

*YSR Template Names*  
rx\_PHA\_YSR\_PublicHousing\_Lease.docx

PHLease

#### **Public Housing Mixed Family Flat Rent Worksheet (MixFam)**

*Top-Level Select Script Files*  
rs\_PHA\_YSR\_PH\_MixFam\_Worksheet\_1.txt

*YSR Script Files*  
rs\_PHA\_YSR\_PH\_MixFam\_Worksheet\_1.txt

MixFam

*YSR Template Names*  
rx\_PHA\_YSR\_PH\_MixFam\_Worksheet.doc

MixFam

#### **QC Inspections (PHABatchQCIInsp\_1)**

*Top-Level Select Script Files*

*YSR Script Files*

rs\_PHA\_YSR\_BatchQCs.txt PHA

*YSR Template Names*

rx\_PHA\_YSR\_BatchQCs.xlsx PHA

**QC Inspections (PHABatchQCInsp)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_YSR\_BatchQCs.txt PHA

*YSR Template Names*

rx\_PHA\_YSR\_BatchQCs.xlsx PHA

**RAD Worksheet (RADWorksheet)***Top-Level Select Script Files*

rs\_PHA\_YSR\_RAD\_Worksheet.txt

*YSR Script Files*

rs\_PHA\_YSR\_RAD\_Worksheet.txt

RADWorks

*YSR Template Names*

rx\_PHA\_YSR\_RAD\_Worksheet.docx

RADWorks

**Rent Comparability Rent Survey (RCUnitS)***Top-Level Select Script Files*

rs\_PHA\_YSR\_RC\_Unit\_Setup\_New.txt

*YSR Script Files*

rs\_PHA\_YSR\_RC\_Unit\_Setup\_New.txt RCUnitS

*YSR Template Names*

rx\_PHA\_YSR\_RC\_Unit\_Setup.xlsx RCUnitS

**Rent Comparability Subject Unit Rent Survey (RCUnitN)***Top-Level Select Script Files*

rs\_PHA\_YSR\_RC\_Unit\_Setup\_1\_New.txt

*YSR Script Files*

rs\_PHA\_YSR\_RC\_Unit\_Setup\_1\_New.txt RCUnitN

*YSR Template Names*

rx\_PHA\_YSR\_RC\_Unit\_Setup\_New.xlsx RCUnitN

**Rent Comparability Subject Unit Rent Survey\_1 (RCUnitN\_1)***Top-Level Select Script Files*

rs\_PHA\_YSR\_RC\_Unit\_Setup\_1\_New\_RFTA.txt

<i>YSR Script Files</i>		
rs_PHA_YSR_RC_Unit_Setup_1_New_RFTA.txt		RCUnitN
<i>YSR Template Names</i>		
rx_PHA_YSR_RC_Unit_Setup_New.xlsx		RCUnitN
<b>Rent Comparability Unit Update Letter (RCUnitU)</b>		
<i>Top-Level Select Script Files</i>		
rs_PHA_YSR_RC_Unit_Update_New.txt		
<i>YSR Script Files</i>		
rs_PHA_YSR_RC_Unit_Update_New.txt		RCUnitU
<i>YSR Template Names</i>		
rx_PHA_YSR_RC_Unit_Update_New.xlsx		RCUnitU
<b>Resident Lease Amendment (TenLeaseAmndmnt1)</b>		
<i>Top-Level Select Script Files</i>		
rs_PHA_YSR_LesLease_Amendment_1.txt		
<i>YSR Script Files</i>		
rs_PHA_YSR_LesLease_Amendment_1.txt		Leaseam
<i>YSR Template Names</i>		
rx_PHA_YSR_LesLease_Amendment.docx		Leaseam
<b>Resident Lease Amendment (TenLeaseAmendmnt)</b>		
<i>Top-Level Select Script Files</i>		
rs_PHA_YSR_LesLease_Amendment.txt		
<i>YSR Script Files</i>		
rs_PHA_YSR_LesLease_Amendment.txt		Leaseam
<i>YSR Template Names</i>		
rx_PHA_YSR_LesLease_Amendment.docx		Leaseam
<b>Resident Recertification Letters (RecertLet)</b>		
<i>Top-Level Select Script Files</i>		
rs_PHA_YSR_Recert.txt		
<i>YSR Script Files</i>		
rs_PHA_YSR_Recert.txt		RecertLe
<i>YSR Template Names</i>		
rx_PHA_YSR_Recert_Tenant.docx		RecertLe
<b>Resident Registration Letter (Resident_Invite)</b>		
<i>Top-Level Select Script Files</i>		
rs_PHA_HC_YSR_Resident_Registration_Letter.txt		
<i>YSR Script Files</i>		

rs\_PHA\_HC\_YSR\_Resident\_Registration\_Letter.txt TenList

*YSR Template Names*  
rx\_PHA\_HC\_YSR\_Resident\_Registration\_Letter.docx TenList

#### SEMAP 10. Rent Calculations (SEMAP10)

*Top-Level Select Script Files*  
rs\_PHA\_YSR\_Rent\_Calc\_Audit.txt

*YSR Script Files*  
rs\_PHA\_YSR\_Rent\_Calc\_Audit.txt RentCalc

*YSR Template Names*  
SELECT CASE '#RptType#' WHEN 'Cross-Tab' THEN 'rx\_PHA\_YSR\_Rent\_Calc\_Audit\_CrossTab.xlsx'  
ELSE 'rx\_PHA\_YSR\_Rent\_Calc\_Audit\_Listing.xlsx' END RentCalc

#### SEMAP 9. Annual Re-examinations Review (SEMAP9)

*Top-Level Select Script Files*

*YSR Script Files*  
rs\_PHA\_YSR\_Recerts\_Annually.txt

AnReExam

*YSR Template Names*  
SELECT CASE '#RptType#' WHEN 'Listing' THEN 'rx\_PHA\_YSR\_Recerts\_Annually\_Listing.xlsx' ELSE  
AnReExam  
'rx\_PHA\_YSR\_Recerts\_Annually\_CrossTab.xlsx' END

#### Supplement to Application for Assisted Housing - HUD 92006

*Top-Level Select Script Files*  
rs\_PHA\_YSR\_SupplementToApplication\_Contact.txt

*YSR Script Files*  
rs\_PHA\_YSR\_SupplementToApplication\_Contact.txt

HUD92006

*YSR Template Names*  
rx\_PHA\_YSR\_SupplementToApplication\_HUD92006.docx  
HUD92006

#### TPA Contract Letter (TPAContr)

*Top-Level Select Script Files*  
rs\_PHA\_YSR\_TPAContract.txt

*YSR Script Files*  
rs\_PHA\_YSR\_TPAContract.txt TPAContr

*YSR Template Names*  
rx\_PHA\_YSR\_TPA\_Contract.doc TPAContr

#### Two Way Referral Letter 052412 (2WayRefr)

*Top-Level Select Script Files*

rs\_PHA\_YSR\_TwoWayReferralLetter.txt

*YSR Script Files*

rs\_PHA\_YSR\_TwoWayReferralLetter.txt

2WayRefr

*YSR Template Names*

rx\_PHA\_YSR\_2\_WAY\_REFERRAL\_LETTER\_052412.docx

2WayRefr

**Two Way Referral Letter ICC 052412 (2WayRICC)***Top-Level Select Script Files*

rs\_PHA\_YSR\_TwoWayICCREferralLetter.txt

*YSR Script Files*

rs\_PHA\_YSR\_TwoWayICCREferralLetter.txt

2WayRICC

*YSR Template Names*

rx\_PHA\_YSR\_2\_WAY\_REFERRAL\_LETTER\_ICC\_052412.docx

2WayRICC

**Unit Offer Print Letter (UnitOfer)***Top-Level Select Script Files*

rs\_PHA\_YSR\_UnitOffer\_UpdateLetter.txt

*YSR Script Files*

rs\_PHA\_YSR\_UnitOffer\_UpdateLetter.txt

UnitOfer

*YSR Template Names*

SELECT CASE '#OfrStatus#' WHEN 'Offered' THEN 'rx\_PHA\_YSR\_UnitOffer\_Offered.docx' WHEN

UnitOfer

'Accepted' THEN 'rx\_PHA\_YSR\_UnitOffer\_Accepted.docx' WHEN 'Canceled' THEN

'rx\_PHA\_YSR\_UnitOffer\_Canceled.docx' WHEN 'Refused' THEN

'rx\_PHA\_YSR\_UnitOffer\_Refused.docx' END

**Unit Offer Print Letter (UnitOfer\_1)***Top-Level Select Script Files*

rs\_PHA\_YSR\_UnitOffer\_UpdateLetter\_1.txt

*YSR Script Files*

rs\_PHA\_YSR\_UnitOffer\_UpdateLetter\_1.txt

UnitOfer

*YSR Template Names*

SELECT CASE '#OfrStatus#' WHEN 'Offered' THEN 'rx\_PHA\_YSR\_UnitOffer\_Offered.docx' WHEN

UnitOfer

'Accepted' THEN 'rx\_PHA\_YSR\_UnitOffer\_Accepted.docx' WHEN 'Canceled' THEN

'rx\_PHA\_YSR\_UnitOffer\_Canceled.docx' WHEN 'Refused' THEN

'rx\_PHA\_YSR\_UnitOffer\_Refused.docx' END

**Vendor/Landlord Registration Letter (Landlord\_Invite)**

<i>Top-Level Select Script Files</i>	
rs_PHA_HC_YSR_Landlord_Registration_Letter.txt	
<i>YSR Script Files</i>	
rs_PHA_HC_YSR_Landlord_Registration_Letter.txt	VendList
<i>YSR Template Names</i>	
rx_PHA_HC_YSR_Landlord_Registration_Letter_NoEmail.docx	VendList
<b>Verification Letters (VerifLTR_1)</b>	
<i>Top-Level Select Script Files</i>	
rs_PHA_YSR_VerifyLetters_Header_1.txt	
<i>YSR Script Files</i>	
rs_PHA_YSR_VerifyLetters_1.txt	Assets
rs_PHA_YSR_VerifyLetters_1.txt	AttCare
rs_PHA_YSR_VerifyLetters_1.txt	AuxCost
rs_PHA_YSR_VerifyLetters_1.txt	BSChild
rs_PHA_YSR_VerifyLetters_1.txt	CertAtCr
rs_PHA_YSR_VerifyLetters_1.txt	ChildSup
rs_PHA_YSR_VerifyLetters_1.txt	Consent
rs_PHA_YSR_VerifyLetters_1.txt	DCCChild
rs_PHA_YSR_VerifyLetters_1.txt	DisAllow
rs_PHA_YSR_VerifyLetters_1.txt	DisReimb
rs_PHA_YSR_VerifyLetters_1.txt	EmpAux
rs_PHA_YSR_VerifyLetters_1.txt	
Employmt	
rs_PHA_YSR_VerifyLetters_1.txt	ExpDef
rs_PHA_YSR_VerifyLetters_1.txt	FTStudnt
rs_PHA_YSR_VerifyLetters_1.txt	
HUD214St	
rs_PHA_YSR_VerifyLetters_1.txt	
HUD9886	
rs_PHA_YSR_VerifyLetters_1.txt	LiveInAD
rs_PHA_YSR_VerifyLetters_1.txt	Medical
rs_PHA_YSR_VerifyLetters_1.txt	MemDef
rs_PHA_YSR_VerifyLetters_1.txt	MilitPay
rs_PHA_YSR_VerifyLetters_1.txt	NChildCr
rs_PHA_YSR_VerifyLetters_1.txt	NonCont
rs_PHA_YSR_VerifyLetters_1.txt	Pension
rs_PHA_YSR_VerifyLetters_1.txt	Prescrip
rs_PHA_YSR_VerifyLetters_1.txt	PubAssis
rs_PHA_YSR_VerifyLetters_1.txt	SelfEmp
rs_PHA_YSR_VerifyLetters_1.txt	SSA_SSI
rs_PHA_YSR_VerifyLetters_1.txt	UABenift
rs_PHA_YSR_VerifyLetters_1.txt	VABenift

*YSR Template Names*

rx_PHA_YSR_Verify_Assets.docx	Assets
rx_PHA_YSR_Verify_AttendantCare.docx	AttCare
rx_PHA_YSR_Verify_AuxiliaryCost.docx	AuxCost
rx_PHA_YSR_Verify_BabySitChildCare.docx	BSChild
rx_PHA_YSR_Verify_CertAttendantCare.docx	CertAtCr
rx_PHA_YSR_Verify_ChildSupport.docx	ChildSup
rx_PHA_YSR_Verify_Consent.docx	Consent
rx_PHA_YSR_Verify_DCCChildCare.docx	DCCChild
rx_PHA_YSR_Verify_DisabilityAllow.docx	DisAllow
rx_PHA_YSR_Verify_DisabilityReimb.docx	DisReimb
rx_PHA_YSR_Verify_EmployerAuxiliary.docx	EmpAux
rx_PHA_YSR_Verify_Employment.docx	
Employmt	
rx_PHA_YSR_Verify_ExpenseDefault.docx	ExpDef
rx_PHA_YSR_Verify_FTStudent.docx	FTStudnt
rx_PHA_YSR_Verify_HUD214Status.docx	
HUD214St	
rx_PHA_YSR_Verify_HUD9886.docx	
HUD9886	
rx_PHA_YSR_Verify_LiveInAide.docx	LiveInAD
rx_PHA_YSR_Verify_Medical.docx	Medical
rx_PHA_YSR_Verify_MemberDefault.docx	MemDef
rx_PHA_YSR_Verify_MilitaryPay.docx	MilitPay
rx_PHA_YSR_Verify_NonReimbChildCare.docx	NChildCr
rx_PHA_YSR_Verify_NonContending.docx	NonCont
rx_PHA_YSR_Verify_Pension.docx	Pension
rx_PHA_YSR_Verify_Prescriptions.docx	Prescrip
rx_PHA_YSR_Verify_PublicAssistance.docx	PubAssis
rx_PHA_YSR_Verify_SelfEmp.docx	SelfEmp
rx_PHA_YSR_Verify_SSA_SSI.docx	SSA_SSI
rx_PHA_YSR_Verify_UABenefits.docx	UABenift
rx_PHA_YSR_Verify_VABenefits.docx	VABenift

**Verification Letters (VerifLTR)***Top-Level Select Script Files*

rs\_PHA\_YSR\_VerifyLetters\_Header.txt

*YSR Script Files*

rs_PHA_YSR_VerifyLetters.txt	Assets
rs_PHA_YSR_VerifyLetters.txt	AttCare
rs_PHA_YSR_VerifyLetters.txt	AuxCost
rs_PHA_YSR_VerifyLetters.txt	BSChild
rs_PHA_YSR_VerifyLetters.txt	CertAtCr

rs_PHA_YSR_VerifyLetters.txt	ChildSup
rs_PHA_YSR_VerifyLetters.txt	Consent
rs_PHA_YSR_VerifyLetters.txt	DCCChild
rs_PHA_YSR_VerifyLetters.txt	Disallow
rs_PHA_YSR_VerifyLetters.txt	DisReimb
rs_PHA_YSR_VerifyLetters.txt	EmpAux
rs_PHA_YSR_VerifyLetters.txt	
Employmt	
rs_PHA_YSR_VerifyLetters.txt	ExpDef
rs_PHA_YSR_VerifyLetters.txt	FTStudnt
rs_PHA_YSR_VerifyLetters.txt	
HUD214St	
rs_PHA_YSR_VerifyLetters.txt	
HUD9886	
rs_PHA_YSR_VerifyLetters.txt	LiveInAD
rs_PHA_YSR_VerifyLetters.txt	Medical
rs_PHA_YSR_VerifyLetters.txt	MemDef
rs_PHA_YSR_VerifyLetters.txt	MilitPay
rs_PHA_YSR_VerifyLetters.txt	NChildCr
rs_PHA_YSR_VerifyLetters.txt	NonCont
rs_PHA_YSR_VerifyLetters.txt	Pension
rs_PHA_YSR_VerifyLetters.txt	Prescrip
rs_PHA_YSR_VerifyLetters.txt	PubAssis
rs_PHA_YSR_VerifyLetters.txt	SelfEmp
rs_PHA_YSR_VerifyLetters.txt	SSA_SSI
rs_PHA_YSR_VerifyLetters.txt	UABenift
rs_PHA_YSR_VerifyLetters.txt	VABenift

*YSR Template Names*

rx_PHA_YSR_Verify_Assets.docx	Assets
rx_PHA_YSR_Verify_AttendantCare.docx	AttCare
rx_PHA_YSR_Verify_AuxiliaryCost.docx	AuxCost
rx_PHA_YSR_Verify_BabySitChildCare.docx	BSChild
rx_PHA_YSR_Verify_CertAttendantCare.docx	CertAtCr
rx_PHA_YSR_Verify_ChildSupport.docx	ChildSup
rx_PHA_YSR_Verify_Consent.docx	Consent
rx_PHA_YSR_Verify_DCCChildCare.docx	DCCChild
rx_PHA_YSR_Verify_DisabilityAllow.docx	Disallow
rx_PHA_YSR_Verify_DisabilityReimb.docx	DisReimb
rx_PHA_YSR_Verify_EmployerAuxiliary.docx	EmpAux
rx_PHA_YSR_Verify_Employment.docx	
Employmt	
rx_PHA_YSR_Verify_ExpenseDefault.docx	ExpDef
rx_PHA_YSR_Verify_FTStudent.docx	FTStudnt

rx_PHA_YSR_Verify_HUD214Status.docx	
HUD214St	
rx_PHA_YSR_Verify_HUD9886.docx	
HUD9886	
rx_PHA_YSR_Verify_LiveInAide.docx	LiveInAD
rx_PHA_YSR_Verify_Medical.docx	Medical
rx_PHA_YSR_Verify_MemberDefault.docx	MemDef
rx_PHA_YSR_Verify_MilitaryPay.docx	MilitPay
rx_PHA_YSR_Verify_NonReimbChildCare.docx	NChildCr
rx_PHA_YSR_Verify_NonContending.docx	NonCont
rx_PHA_YSR_Verify_Pension.docx	Pension
rx_PHA_YSR_Verify_Prescriptions.docx	Prescrip
rx_PHA_YSR_Verify_PublicAssistance.docx	PubAssis
rx_PHA_YSR_Verify_SelfEmp.docx	SelfEmp
rx_PHA_YSR_Verify_SSA_SSI.docx	SSA_SSI
rx_PHA_YSR_Verify_UABenefits.docx	UABenift
rx_PHA_YSR_Verify_VABenefits.docx	VABenift

#### Verified Items (Criteria - Admission or Reexamination) (SEMAP31)

##### *Top-Level Select Script Files*

rs\_PHA\_YSR\_PHA\_Verified\_At\_Admission.txt

##### *YSR Script Files*

rs\_PHA\_YSR\_PHA\_Verified\_At\_Admission.txt

VerType

##### *YSR Template Names*

```
SELECT CASE '#RptType#' WHEN 'Cross-Tab' THEN
      'rx_PHA_YSR_PHA_Verified_At_Admission_CrossTab.xlsx' WHEN 'Graphical' THEN
      'rx_PHA_YSR_PHA_Verified_At_Admission_Graphical.xlsx' ELSE
      'rx_PHA_YSR_PHA_Verified_At_Admission_Listing.xlsx' END
```

VerType

#### VMS - GL Reconciliation Report (VMSGRL)

##### *Top-Level Select Script Files*

##### *YSR Script Files*

rs\_PHA\_YSR\_VMS\_GL\_Reconciliation\_Report.txt

VMSGRL

##### *YSR Template Names*

rx\_PHA\_YSR\_VMS\_GL\_Reconciliation\_Report.xlsx

VMSGRL

#### VMS Comparison Report (VMSCom)

##### *Top-Level Select Script Files*

##### *YSR Script Files*

rs\_PHA\_YSR\_VMS\_Comparison.txt

VMSCom

##### *YSR Template Names*

rx\_PHA\_YSR\_VMS\_Comparison.xlsx

VMSCom

**VMS Monthly Report (VMSSum)***Top-Level Select Script Files*

rs\_PHA\_YSR\_VMS.txt

*YSR Script Files*

rs\_PHA\_YSR\_VMS.txt

VMS

*YSR Template Names*

Select CASE '#RptType#' WHEN 'Summary' THEN 'rx\_PHA\_YSR\_VMS\_Summary.xlsx' ELSE

VMS

'rx\_PHA\_YSR\_VMS.xlsx' END

**VMS Months to be Adjusted (VMSMA)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_YSR\_VMS\_MonthsToAdjust.txt

VMSMA

*YSR Template Names*

rx\_PHA\_YSR\_VMS\_MonthsToAdjust.xlsx

VMSMA

**Voucher Utilization (VoucherU)***Top-Level Select Script Files*

rs\_PHA\_YSR\_VoucherUtilization.txt

*YSR Script Files*

rs\_PHA\_YSR\_VoucherUtilization.txt

VoucherS

*YSR Template Names*

rx\_PHA\_YSR\_VoucherUtilization.xlsx

VoucherS

**Vouchers Outstanding (vouout)***Top-Level Select Script Files**YSR Script Files*

rs\_PHA\_Voucher\_Outstanding\_YSR.txt

Vou\_Out

*YSR Template Names*

rx\_PHA\_Vouchers\_Outstanding\_YSR.xlsx

Vou\_Out

**Waiting List History (WLHist)***Top-Level Select Script Files*

rs\_WL\_YSR\_WLHistory.txt

*YSR Script Files*

rs\_WL\_YSR\_WLHistory.txt

WL

*YSR Template Names*

rx\_WL\_YSR\_WLHistory.xlsx

WL

## RentCafe

### RentCafe Condo Board Member Reports (RentCafeCondo)

*Top-Level Select Script Files*

*YSR Script Files*

YSR\_CondoOwnerList.txt  
YSR\_CondoWorkOrders.txt

CondoOL

CondoWO

*YSR Template Names*

Select CASE WHEN 'Aged Receivable' In (#sReport#) then 'YSR_CondoAR.xlsx' ELSE 'YSR_Condo_Bank.xlsx' end	CondoAR
Select CASE WHEN 'Owner List' In (#sReport#) then 'YSR_CondoOwnerList.xlsx' ELSE 'YSR_Condo_Bank.xlsx' end	CondoOL
Select CASE WHEN 'Work Orders' In (#sReport#) then 'YSR_CondoWorkOrders.xlsx' ELSE CondoWO 'YSR_Condo_Bank.xlsx' end	

### RentCafeCondo Aged Receivable Report (RentCafeCondoAR)

*Top-Level Select Script Files*

*YSR Script Files*

*YSR Template Names*

YSR_CondoAR.xlsx	CondoAR
------------------	---------

### RentCafeCondo Owner List Report (RentCafeCondoOL)

*Top-Level Select Script Files*

*YSR Script Files*

YSR_CondoOwnerList.txt	CondoOL
------------------------	---------

*YSR Template Names*

YSR_CondoOwnerList.xlsx	CondoOL
-------------------------	---------

## Renters Insurance

### RI-Compliance - Non-compliant Resident (ComplianceReside)

*Top-Level Select Script Files*

rs\_ITF\_RI\_ICompliance\_Non-Compliant\_Resident\_Filter.txt

*YSR Script Files*

rs_ITF_RI_ICompliance_Non-Compliant_Resident_Script.txt	ExprdPol
---	----------

*YSR Template Names*

yDoc_Renters_Insurance_Compliance_Non-Compliant_Resident.docx	ExprdPol
---	----------

**RI-Compliance - Transfer Notice (Change Address)***Top-Level Select Script Files*

rs\_ITF\_RI\_Compliance\_Transfer\_Note\_Filter.txt

*YSR Script Files*

rs\_ITF\_RI\_Compliance\_Transfer\_Note\_Script.txt

ExprdPol

*YSR Template Names*

yDoc\_Renters\_Insurance\_Compliance\_Transfer\_Note\_Change\_Address.docx

ExprdPol

**RI-Expired Policy Notification (ExprdPolicyNotic)***Top-Level Select Script Files*

rs\_ITF\_RI\_Expired\_Policy\_Notification\_Filter.txt

*YSR Script Files*

rs\_ITF\_RI\_Expired\_Policy\_Notification\_Script.txt

ExprdPol

*YSR Template Names*

yDoc\_Renters\_Insurance\_Expired\_Policy\_Notification.docx

ExprdPol

**RI-Expiring Policy Notification (ExprnPolicyNotic)***Top-Level Select Script Files*

rs\_ITF\_RI\_Expiring\_Policy\_Notification\_Filter.txt

*YSR Script Files*

rs\_ITF\_RI\_Expiring\_Policy\_Notification\_Script.txt

ExprdPol

*YSR Template Names*

yDoc\_Renters\_Insurance\_Expiring\_Policy\_Notification.docx

ExprdPol

**RI-Introduction Notice (IntroNotice)***Top-Level Select Script Files*

rs\_ITF\_RI\_Introduction\_Note\_Filter.txt

*YSR Script Files*

rs\_ITF\_RI\_Introduction\_Note\_Script.txt

ExprdPol

*YSR Template Names*

yDoc\_Renters\_Insurance\_Introduction\_Note\_Mandatory.docx

ExprdPol

**RI-Introduction Notice (Voluntary) (IntroNtceVoluntr)***Top-Level Select Script Files*

rs\_ITF\_RI\_Introduction\_Note\_Voluntary\_Filter.txt

*YSR Script Files*

rs\_ITF\_RI\_Introduction\_Note\_Voluntary\_Script.txt

ExprdPol

*YSR Template Names*

yDoc\_Renters\_Insurance\_Introduction\_Note\_Voluntary.docx

ExprdPol

**RI-Master Policy - Auto-Enroll Notification (MasterPoliEnroll)**

*Top-Level Select Script Files*

rs\_ITF\_RI\_Master\_Policy\_Auto\_Enroll\_Notification\_Filter.txt

*YSR Script Files*

rs\_ITF\_RI\_Master\_Policy\_Auto\_Enroll\_Notification\_Script.txt

ExprdPol

*YSR Template Names*

yDoc\_Master\_Policy\_Auto\_Enroll\_Notification.docx

ExprdPol

**RI-Renewals Notice (RenewalsNotice)***Top-Level Select Script Files*

rs\_ITF\_RI\_Renters\_Insurance\_Renewals\_Note\_Filter.txt

*YSR Script Files*

rs\_ITF\_RI\_Renters\_Insurance\_Renewals\_Note\_Script.txt

ExprdPol

*YSR Template Names*

yDoc\_Renters\_Insurance\_Renewals\_Note.docx

ExprdPol

**RI-Third Party Policy Notice (ThrdPtyPolicyNot)***Top-Level Select Script Files*

rs\_ITF\_RI\_Third\_Party\_Policy\_Note\_Filter.txt

*YSR Script Files*

rs\_ITF\_RI\_Third\_Party\_Policy\_Note\_Script.txt

ExprdPol

*YSR Template Names*

yDoc\_Third\_Party\_Policy\_Note.docx

ExprdPol

## Residential

**Residential Resident Activity (ResActivity)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

ysr\_ResResidentActivity.xlsx

resactvt

**Residential Traffic Detail (ResTrafficDetail)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

ysr\_ResTrafficDetailReport.xlsx

TraffDet

**Residential Unit Availability (ResUnitAvlbl)***Top-Level Select Script Files*

*YSR Script Files**YSR Template Names*

ysr\_ResUnitAvailability.xlsx

unitavbl

## Section 8

**HCV Waiting List Placement Letter (WL\_S8Select)***Top-Level Select Script Files*

rs\_WL\_YSR\_Section8\_Waitlist\_Selection.txt

*YSR Script Files*

rs\_WL\_YSR\_Section8\_Waitlist\_Selection.txt

TenList

*YSR Template Names*

rx\_WL\_YSR\_Section8\_Waitlist\_Selection.docx

TenList

**HCV Waitlist Non Selection (WL\_S8NoSelect)***Top-Level Select Script Files*

rs\_WL\_YSR\_Section8\_Waitlist\_NonSelection.txt

*YSR Script Files*

rs\_WL\_YSR\_Section8\_Waitlist\_NonSelection.txt

TenList

*YSR Template Names*

rx\_WL\_YSR\_Section8\_Waitlist\_NonSelection.docx

TenList

**Waiting List New Applicants Letters (WLNewApps)***Top-Level Select Script Files*

rs\_WL\_YSR\_Waiting\_List\_Letters\_NewApp.txt

*YSR Script Files*

rs\_WL\_YSR\_Waiting\_List\_Letters\_NewApp.txt

ConLet

*YSR Template Names*

Select CASE '#RptType#' WHEN 'Confirmation Letter' THEN

ConLet

'rx\_WL\_YSR\_Waiting\_List\_Letters\_NewApp\_ConfLet.docx' ELSE

'rx\_WL\_YSR\_Waiting\_List\_Letters\_NewApp.docx' END

## Senior Housing

**Active Clients (EHRHHActClnts)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRXActiveClient.txt

d

<i>YSR Template Names</i>		
rs_SeniorEHRXActiveClient.xlsx	d	
<b>Allergy Audit Report (EHRAllergyAudit)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXAllergyAuditNew.txt	Detail	
<i>YSR Template Names</i>		
rs_SeniorEHRXAllergyAuditNew.xlsx	Detail	
<b>Allergy Report (EHRAllergy)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXAllergyNew.txt	Detail	
<i>YSR Template Names</i>		
rs_SeniorEHRXAllergyNew.xlsx	Detail	
<b>Assessment &amp; Service Result Report (EHRAssSerRes)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXAssessmentResultServiceNew.txt	SerRep	
<i>YSR Template Names</i>		
rs_SeniorEHRXAssessmentResultServiceNew.xlsx	SerRep	
<b>Assessment Library Conditional Logic Report (EHRAssLibConLogi)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXAssessmentLibraryConditionalLogic.txt	ALCL	
<i>YSR Template Names</i>		
rs_SeniorEHRXAssessmentLibraryConditionalLogic.xlsx	ALCL	
<b>Assessment Result Report (EHRAssRes)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXAssessmentResultNew.txt	AssRep	
<i>YSR Template Names</i>		
rs_SeniorEHRXAssessmentResultNew.xlsx	AssRep	
<b>Assessments Due (EHRHHAssessDue)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		

rs_SeniorEHRXAssessmentsDue.txt	d
<i>YSR Template Names</i>	
rs_SeniorEHRXAssessmentsDue.xlsx	d
<b>Billable Hours by Payer (EHRHHBillHrs)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_SeniorEHRxBillableHoursByPayer.txt	BillHrs
<i>YSR Template Names</i>	
rs_SeniorEHRxBillableHoursByPayer.xlsx	BillHrs
<b>Billable Visits (EHRHHBillVis)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_SeniorEHRxBillableVisits.txt	BillVis
<i>YSR Template Names</i>	
rs_SeniorEHRxBillableVisits.xlsx	BillVis
<b>CAA Audit Report (EHRMDSAUDIT)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_SeniorEHRXMDSAuditReport.txt	detail
<i>YSR Template Names</i>	
rs_SeniorEHRXMDSAuditReport.xlsx	detail
<b>CAA Reports (EHRxCAResSum)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_SeniorEHRXMDSActivities.txt	
EHRMDSA	
rs_SeniorEHRXMDSBehavioralSymptoms.txt	
EHRMDSBS	
rs_SeniorEHRXMDSCognitiveLoss.txt	
EHRMDSCL	
rs_SeniorEHRXMDSCommunication.txt	
EHRMDSCO	
rs_SeniorEHRXMDSDehydrationMaintenance.txt	
EHRMDSDM	
rs_SeniorEHRXMDSDelirium.txt	
EHRMDEL	
rs_SeniorEHRXMDSFalls.txt	
EHRMDSF	

- rs\_SeniorEHRXMDSTestingTubes.txt
- EHRMDSFT
  - rs\_SeniorEHRXMDSTestingFunctionalStatus.txt
- EHRMDSFS
  - rs\_SeniorEHRXMDSMoodState.txt
- EHRMDSMS
  - rs\_SeniorEHRXMDSNutritionalStatus.TXT
- EHRMDSNS
  - rs\_SeniorEHRXMDSOralDentalConditionProblem.txt
- EHRMDSDP
  - rs\_SeniorEHRXMDSPain.txt
- EHRMDSPA
  - rs\_SeniorEHRXMDSPhysicalRestraints.TXT
- EHRMDSPR
  - rs\_SeniorEHRXMDSPressureUlcers.txt
- EHRMDSPU
  - rs\_SeniorEHRXMDSPychosocialWellBeing.txt
- EHRMDSPW
  - rs\_SeniorEHRXMDSSReturnToCommunity.txt
- EHRMDSRC
  - rs\_SeniorEHRXMDSSReviewOfIndicatorsOfPsychotropicDrugUse.txt
- EHRMDSPD
  - rs\_SeniorEHRXMDSUrinaryIncontinence.txt
- EHRMDSUI
  - rs\_SeniorEHRXMDSSVisualFunction.txt
- EHRMDSVF

*YSR Template Names*

- rs\_SeniorEHRXMDSDelirium.xlsx
- EHRMDEL
  - rs\_SeniorEHRXMDSActivities.xlsx
- EHRMDSA
  - rs\_SeniorEHRXMDSBehavioralSymptoms.xlsx
- EHRMDSBS
  - rs\_SeniorEHRXMDSCognitiveLoss.xlsx
- EHRMDSCL
  - rs\_SeniorEHRXMDSCommunication.xlsx
- EHRMDSCO
  - rs\_SeniorEHRXMDSDehydrationMaintenance.xlsx
- EHRMDSDM
  - rs\_SeniorEHRXMDSOralDentalConditionProblem.xlsx
- EHRMDSDP
  - rs\_SeniorEHRXMDSFalls.xlsx
- EHRMDSF
  - rs\_SeniorEHRXMDSFunctionalStatus.xlsx

EHRMDSFS  
 rs\_SeniorEHRXMDSTubes.xlsx

EHRMDSFT  
 rs\_SeniorEHRXMDSMoodState.xlsx

EHRMDSMS  
 rs\_SeniorEHRXMDSNutritionalStatus.xlsx

EHRMDSNS  
 rs\_SeniorEHRXMDSPain.xlsx

EHRMDSPA  
 rs\_SeniorEHRXMDSSReviewOfIndicatorsOfPsychotropicDrugUse.xlsx

EHRMDSPD  
 rs\_SeniorEHRXMDSPhysicalRestraints.xlsx

EHRMDSPR  
 rs\_SeniorEHRXMDSPressureUlcers.xlsx

EHRMDSPU  
 rs\_SeniorEHRXMDSPychosocialWellBeing.xlsx

EHRMDSPW  
 rs\_SeniorEHRXMDSSReturnToCommunity.xlsx

EHRMDSRC  
 rs\_SeniorEHRXMDSUrinaryIncontinence.xlsx

EHRMDSUI  
 rs\_SeniorEHRXMDSSVisualFunction.xlsx

EHRMDSVF

**Care Action Details (CareActionDetail)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorCarePlanDetails.txt Detail

*YSR Template Names*

rs\_SeniorCarePlanDetails.xlsx Detail

**Care Plan Report (HHCare)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRxCarePlanHH.txt HHCare

*YSR Template Names*

rs\_SeniorEHRxCarePlanHH.xlsx HHCare

**Census By Payer (EHRHHCensusPay)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRxCensusByPayer.txt CensusPa

*YSR Template Names*

rs_SeniorEHRxCensusByPayer.xlsx	CensusPa
<b>Census Residents Including Leave of Absence (EHRCensusResLOA)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_SeniorEHRXCensusResidentLOA.txt	CRLOA
<i>YSR Template Names</i>	
rs_SeniorEHRXCensusResidentLOA.xlsx	CRLOA
<b>Cognitive Performance Scale (EHRCPS)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_SeniorEHRXCPSReport.txt	Detail
<i>YSR Template Names</i>	
rs_SeniorEHRXCPSReport.xlsx	Detail
<b>Daily Task Log Manual Form (EHRXDTL)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_SeniorEHRXDailyTeamAssignment.txt	SDLR
<i>YSR Template Names</i>	
rs_SeniorEHRXDailyTeamAssignment.xlsx	SDLR
<b>Daily Task Log Manual Form Link (EHRXDTLLink)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_SeniorEHRXDailyTeamAssignmentHomeHealth.txt	DTLLink
<i>YSR Template Names</i>	
rs_SeniorEHRXDailyTeamAssignmentHomeHealth.xlsx	DTLLink
<b>Description of EHRCMS672 (EHRCMS672)</b>	
<i>Top-Level Select Script Files</i>	
rs_SeniorEHRCMS672.txt	
<i>YSR Script Files</i>	
rs_SeniorEHRCMS672.txt	rpt
<i>YSR Template Names</i>	
rs_SeniorEHRCMS672.xlsx	rpt
<b>Description of EHRCMS672Supp (EHRCMS672Supp)</b>	
<i>Top-Level Select Script Files</i>	
rs_SeniorEHRCMS672Supp.txt	

<i>YSR Script Files</i>		
rs_SeniorEHRCMS672Supp.txt		rpt
<i>YSR Template Names</i>		
rs_SeniorEHRCMS672Supp.xlsx		rpt
<b>Description of EHRxAuditReport (EHRxAuditReport)</b>		
<i>Top-Level Select Script Files</i>		
rs_SeniorEHRxAuditReport.txt		
<i>YSR Script Files</i>		
rs_SeniorEHRxAuditReport.txt		rpt
<i>YSR Template Names</i>		
rs_SeniorEHRxAuditReport.xlsx		rpt
<b>Description of EHRxCAAMDSSum (EHRxCAAMDSSum)</b>		
<i>Top-Level Select Script Files</i>		
rs_SeniorEHRxCAAMDS.txt		
<i>YSR Script Files</i>		
rs_SeniorEHRxCAAMDS.txt		rpt
<i>YSR Template Names</i>		
rs_SeniorEHRxCAAMDS.xlsx		rpt
<b>Description of EHRxSectionS_AR (EHRxSectionS_AR)</b>		
<i>Top-Level Select Script Files</i>		
rs_SeniorEHRxSectionS_AR.txt		
<i>YSR Script Files</i>		
rs_SeniorEHRxSectionS_AR.txt		rpt
<i>YSR Template Names</i>		
rs_SeniorEHRxSectionS_AR.xlsx		rpt
<b>Description of EHRxSectionS_CA (EHRxSectionS_CA)</b>		
<i>Top-Level Select Script Files</i>		
rs_SeniorEHRxSectionS_CA.txt		
<i>YSR Script Files</i>		
rs_SeniorEHRxSectionS_CA.txt		rpt
<i>YSR Template Names</i>		
rs_SeniorEHRxSectionS_CA.xlsx		rpt
<b>Description of EHRxSectionS_FL (EHRxSectionS_FL)</b>		
<i>Top-Level Select Script Files</i>		
rs_SeniorEHRxSectionS_FL.txt		
<i>YSR Script Files</i>		

rs\_SeniorEHRxSectionS\_FL.txt rpt

*YSR Template Names*

rs\_SeniorEHRxSectionS\_FL.xlsx rpt

#### **Description of EHRxSectionS\_IL (EHRxSectionS\_IL)**

*Top-Level Select Script Files*

rs\_SeniorEHRxSectionS\_IL.txt

*YSR Script Files*

rs\_SeniorEHRxSectionS\_IL.txt

rpt

*YSR Template Names*

rs\_SeniorEHRxSectionS\_IL.xlsx

rpt

#### **Description of EHRxSectionS\_LA (EHRxSectionS\_LA)**

*Top-Level Select Script Files*

rs\_SeniorEHRxSectionS\_LA.txt

*YSR Script Files*

rs\_SeniorEHRxSectionS\_LA.txt

rpt

*YSR Template Names*

rs\_SeniorEHRxSectionS\_LA.xlsx

rpt

#### **Description of EHRxSectionS\_MA (EHRxSectionS\_MA)**

*Top-Level Select Script Files*

rs\_SeniorEHRxSectionS\_MA.txt

*YSR Script Files*

rs\_SeniorEHRxSectionS\_MA.txt

rpt

*YSR Template Names*

rs\_SeniorEHRxSectionS\_MA.xlsx

rpt

#### **Description of EHRxSectionS\_MD (EHRxSectionS\_MD)**

*Top-Level Select Script Files*

rs\_SeniorEHRxSectionS\_MD.txt

*YSR Script Files*

rs\_SeniorEHRxSectionS\_MD.txt

rpt

*YSR Template Names*

rs\_SeniorEHRxSectionS\_MD.xlsx

rpt

#### **Description of EHRxSectionS\_ME (EHRxSectionS\_ME)**

*Top-Level Select Script Files*

rs\_SeniorEHRxSectionS\_ME.txt

*YSR Script Files*

rs\_SeniorEHRxSectionS\_ME.txt

rpt

<i>YSR Template Names</i>		
rs_SeniorEHRxSectionS_ME.xlsx		rpt
<b>Description of EHRxSectionS_MS (EHRxSectionS_MS)</b>		
<i>Top-Level Select Script Files</i>		
rs_SeniorEHRxSectionS_MS.txt		
<i>YSR Script Files</i>		
rs_SeniorEHRxSectionS_MS.txt		rpt
<i>YSR Template Names</i>		
rs_SeniorEHRxSectionS_MS.xlsx		rpt
<b>Description of EHRxSectionS_ND (EHRxSectionS_ND)</b>		
<i>Top-Level Select Script Files</i>		
rs_SeniorEHRxSectionS_ND.txt		
<i>YSR Script Files</i>		
rs_SeniorEHRxSectionS_ND.txt		rpt
<i>YSR Template Names</i>		
rs_SeniorEHRxSectionS_ND.xlsx		rpt
<b>Description of EHRxSectionS_NE (EHRxSectionS_NE)</b>		
<i>Top-Level Select Script Files</i>		
rs_SeniorEHRxSectionS_NE.txt		
<i>YSR Script Files</i>		
rs_SeniorEHRxSectionS_NE.txt		rpt
<i>YSR Template Names</i>		
rs_SeniorEHRxSectionS_NE.xlsx		rpt
<b>Description of EHRxSectionS_NY (EHRxSectionS_NY)</b>		
<i>Top-Level Select Script Files</i>		
rs_SeniorEHRxSectionS_NY.txt		
<i>YSR Script Files</i>		
rs_SeniorEHRxSectionS_NY.txt		rpt
<i>YSR Template Names</i>		
rs_SeniorEHRxSectionS_NY.xlsx		rpt
<b>Description of EHRxSectionS_OH (EHRxSectionS_OH)</b>		
<i>Top-Level Select Script Files</i>		
rs_SeniorEHRxSectionS_OH.txt		
<i>YSR Script Files</i>		
rs_SeniorEHRxSectionS_OH.txt		rpt
<i>YSR Template Names</i>		

rs\_SeniorEHRxSectionS\_OH.xlsx rpt

#### Description of EHRxSectionS\_PA (EHRxSectionS\_PA)

*Top-Level Select Script Files*

rs\_SeniorEHRxSectionS\_PA.txt

*YSR Script Files*

rs\_SeniorEHRxSectionS\_PA.txt

rpt

*YSR Template Names*

rs\_SeniorEHRxSectionS\_PA.xlsx

rpt

#### Description of EHRxSectionS\_SD (EHRxSectionS\_SD)

*Top-Level Select Script Files*

rs\_SeniorEHRxSectionS\_SD.txt

*YSR Script Files*

rs\_SeniorEHRxSectionS\_SD.txt

rpt

*YSR Template Names*

rs\_SeniorEHRxSectionS\_SD.xlsx

rpt

#### Description of EHRxSectionS\_VA (EHRxSectionS\_VA)

*Top-Level Select Script Files*

rs\_SeniorEHRxSectionS\_VA.txt

*YSR Script Files*

rs\_SeniorEHRxSectionS\_VA.txt

rpt

*YSR Template Names*

rs\_SeniorEHRxSectionS\_VA.xlsx

rpt

#### Description of EHRxSectionS\_VT (EHRxSectionS\_VT)

*Top-Level Select Script Files*

rs\_SeniorEHRxSectionS\_VT.txt

*YSR Script Files*

rs\_SeniorEHRxSectionS\_VT.txt

rpt

*YSR Template Names*

rs\_SeniorEHRxSectionS\_VT.xlsx

rpt

#### Description of EHRxSectionS\_WV (EHRxSectionS\_WV)

*Top-Level Select Script Files*

rs\_SeniorEHRxSectionS\_WV.txt

*YSR Script Files*

rs\_SeniorEHRxSectionS\_WV.txt

rpt

*YSR Template Names*

rs\_SeniorEHRxSectionS\_WV.xlsx

rpt

**Description of EHRXSWR (EHRXSWR)***Top-Level Select Script Files*

rs\_SeniorEHRXStopAndWatchReport.txt

*YSR Script Files*

rs\_SeniorEHRXStopAndWatchReport.txt

rpt

*YSR Template Names*

rs\_SeniorEHRXStopAndWatchReport.xlsx

rpt

**Description of FacilityMatrix (FacilityMatrix)***Top-Level Select Script Files*

rs\_SeniorEHRFacilityMatrix.txt

*YSR Script Files*

rs\_SeniorEHRFacilityMatrix.txt

rpt

*YSR Template Names*

rs\_SeniorEHRFacilityMatrix.xlsx

rpt

**Description of MDS30Trans (MDS30Trans)***Top-Level Select Script Files*

rs\_SeniorMDS30Transmission.txt

*YSR Script Files*

rs\_SeniorMDS30Transmission.txt

rpt

*YSR Template Names*

rs\_SeniorMDS30Transmission.xlsx

rpt

**Description of MDSCATrans (MDSCATrans)***Top-Level Select Script Files*

rs\_SeniorMDSCATransmission.txt

*YSR Script Files*

rs\_SeniorMDSCATransmission.txt

rpt

*YSR Template Names*

rs\_SeniorMDSCATransmission.xlsx

rpt

**Description of MDSWorkNote (MDSWorkNote)***Top-Level Select Script Files*

rs\_SeniorEHRMDSWorksheetNote.txt

*YSR Script Files*

rs\_SeniorEHRMDSWorksheetNote.txt

rpt

*YSR Template Names*

rs\_SeniorEHRMDSWorksheetNote.xlsx

rpt

**Description of NY10-01-2018 (NY10-01-2018)**

*Top-Level Select Script Files*

rs\_SeniorEHRxSectionS\_NY10-01-2018.txt

*YSR Script Files*

rs\_SeniorEHRxSectionS\_NY10-01-2018.txt

rpt

*YSR Template Names*

rs\_SeniorEHRxSectionS\_NY10-01-2018.xlsx

rpt

**Discharged Clients (EHRHHDischrCli)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRXDischargedClients.txt

d

*YSR Template Names*

rs\_SeniorEHRXDischargedClients.xlsx

d

**Drug ID Update Report (EHRDrugIdUpdate)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRXDrugIDUpdateReport.txt

DIU

*YSR Template Names*

rs\_SeniorEHRXDrugIDUpdateReport.xlsx

DIU

**EHR Pharmacy Order Review Report (EHRPharmRev)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRxPharmacyOrderReview.txt

EPR

*YSR Template Names*

rs\_SeniorEHRxPharmacyOrderReview.xlsx

EPR

**EHRx Home Health Appointments Report: Caregiver***Top-Level Select Script Files*

rs\_SeniorEHRxHomeHealth\_MasterSchedulerCaregiver.txt

*YSR Script Files*

rs\_SeniorEHRxHomeHealth\_MasterSchedulerCaregiver.txt

APPTSCR

*YSR Template Names*

rs\_SeniorEHRxHomeHealth\_MasterSchedulerCaregiver.xlsx

APPTSCR

**EHRx Home Health Appointments Report: Client***Top-Level Select Script Files*

rs\_SeniorEHRxHomeHealth\_MasterSchedulerClient.txt

*YSR Script Files*

rs_SeniorEHRxHomeHealth_MasterSchedulerClient.txt	APPTSCL
<i>YSR Template Names</i>	
rs_SeniorEHRxHomeHealth_MasterSchedulerClient.xlsx	APPTSCL
<b>EHRx Home Health Coverage Area Log Report (EHRHHCovArea)</b>	
<i>Top-Level Select Script Files</i>	
rs_SeniorEHRxHomeHealth_CoverageArea.txt	
<i>YSR Script Files</i>	
rs_SeniorEHRxHomeHealth_CoverageArea.txt	areas
<i>YSR Template Names</i>	
rs_SeniorEHRxHomeHealth_CoverageArea.xlsx	areas
<b>EHRx Home Health User Wage Reconciliation Log Report</b>	
<i>Top-Level Select Script Files</i>	
rs_SeniorEHRxHomeHealth_WageReconciliation.txt	
<i>YSR Script Files</i>	
rs_SeniorEHRxHomeHealth_WageReconciliation.txt	wages
<i>YSR Template Names</i>	
rs_SeniorEHRxHomeHealth_WageReconciliation.xlsx	wages
<b>EHRx Home Health User Wage Report (EHRHHWage)</b>	
<i>Top-Level Select Script Files</i>	
rs_SeniorEHRxHomeHealth_UserWage.txt	
<i>YSR Script Files</i>	
rs_SeniorEHRxHomeHealth_UserWage.txt	wages
<i>YSR Template Names</i>	
rs_SeniorEHRxHomeHealth_UserWage.xlsx	wages
<b>Incident Tracking Form (EHRxIncTrackForm)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
rs_SeniorEHRxIncidentTrackingForm.txt	EITF
rs_SeniorEHRxIncidentTrackingForm.txt	EITFSF
rs_SeniorEHRxIncidentTrackingForm.txt	EITFSO
<i>YSR Template Names</i>	
SeniorEHRxIncidentTrackingForm.xlsx	EITF
SeniorEHRxIncidentTrackingFormDSF.xlsx	EITFSF
SeniorEHRxIncidentTrackingFormDSO.xlsx	EITFSO
<b>Incomplete MDS Assessments (EHRMDSAssess)</b>	
<i>Top-Level Select Script Files</i>	

<i>YSR Script Files</i>		
rs_SeniorEHRXIncompleteMDSSessments.txt		Detail
<i>YSR Template Names</i>		
rs_SeniorEHRXIncompleteMDSSessments.xlsx		Detail
<b>MDS Log Report (EHRMDSLog)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXMDSLogReport.txt		Detail
<i>YSR Template Names</i>		
rs_SeniorEHRXMDSLogReport.xlsx		Detail
<b>MDS Report (EHRxMDS)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXMDSReport.txt		MDS
<i>YSR Template Names</i>		
rs_SeniorEHRXMDSReport.xlsx		MDS
<b>MDS Transmitted Forms Grouped and Sorted by Batch ID</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXMDSTransmittedForm.txt		
MDSTrans		
<i>YSR Template Names</i>		
rs_SeniorEHRXMDSTransmittedForm.xlsx		
MDSTrans		
<b>MDS Worksheet Report (EHRMDSWorksheet)</b>		
<i>Top-Level Select Script Files</i>		
rs_SeniorEHRMDSWorksheet.txt		
<i>YSR Script Files</i>		
rs_SeniorEHRMDSWorksheet.txt		
MDSWorks		
<i>YSR Template Names</i>		
rs_SeniorEHRMDSWorksheet.xlsx		
MDSWorks		
<b>MDSCA Assessment CAP/RAP Report (EHRMDSCAAssCRpt)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		

rs\_SeniorEHRXMDSCAAssesmentCAPRAP.txt d

*YSR Template Names*

rs\_SeniorEHRXMDSCAAssesmentCAPRAP.xlsx d

### Message Center Audit (EHRXMSG)

*Top-Level Select Script Files*

*YSR Script Files*

rs\_SeniorEHRXMessageCenter.txt

EHRXMSG

*YSR Template Names*

rs\_SeniorEHRXMessageCenter.xlsx

EHRXMSG

### Missed Visit assessment report (MiVisit)

*Top-Level Select Script Files*

*YSR Script Files*

rs\_SeniorEHRxMissVisitAssessment.txt MiVisit

*YSR Template Names*

rs\_SeniorEHRxMissVisitAssessment.xlsx MiVisit

### Missed Visit Log (MiViLog)

*Top-Level Select Script Files*

*YSR Script Files*

rs\_SeniorEHRxMissVisitAssessmentLog.txt MiViLog

*YSR Template Names*

rs\_SeniorEHRxMissVisitAssessmentLog.xlsx MiViLog

### Monthly Task Log (EHRMTL)

*Top-Level Select Script Files*

rs\_SeniorEHRXMonthlyTaskLogYSR.txt

*YSR Script Files*

rs\_SeniorEHRXMonthlyTaskLog.txt MTI

*YSR Template Names*

rs\_SeniorEHRXMonthlyTaskLog.xlsx MTI

### Monthly Task Log (EHRMTLLink)

*Top-Level Select Script Files*

rs\_SeniorEHRXMonthlyTaskLogYSR.txt

*YSR Script Files*

rs\_SeniorEHRXMonthlyTaskLog.txt MTI

<i>YSR Template Names</i>		
rs_SeniorEHRXMonthlyTaskLog.xlsx		MTI
<b>Non Admissions (EHRHHNonAdms)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXNonAdmission.txt		d
<i>YSR Template Names</i>		
rs_SeniorEHRXNonAdmissions.xlsx		d
<b>Non English Speaking/Other Primary Language Residents</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXResNonEngLang.txt		Language
<i>YSR Template Names</i>		
rs_SeniorEHRXResNonEngLang.xlsx		Language
<b>Non-medical Comprehensive assessment (HH_Comp)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRxComprehensiveAssessment.txt		
HH_Comp		
<i>YSR Template Names</i>		
rs_SeniorEHRxComprehensiveAssessment.xlsx		
HH_Comp		
<b>Order and Resident Census Change Report (EHRxOrderCng)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXOrderChange.txt		EHRxOrd
<i>YSR Template Names</i>		
rs_SeniorEHRXOrderChange.xlsx		EHRxOrd
<b>Payroll History (EHRHHPayroll)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXPayRollHistory.txt		d
<i>YSR Template Names</i>		
rs_SeniorEHRXPayRollHistory.xlsx		d
<b>Physician List (EHRHHPhyList)</b>		

*Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRXPhysiciansList.txt

d

*YSR Template Names*

rs\_SeniorEHRXPhysiciansList.xlsx

d

**Resident Behavior Detail Report (EHRResBehDetail)***Top-Level Select Script Files*

rs\_SeniorEHRXResidentBehaviorDetailYSR.txt

*YSR Script Files*

rs\_SeniorEHRXResidentBehaviorDetail.txt

ResBehav

*YSR Template Names*

rs\_SeniorEHRXResidentBehaviorDetail.xlsx

ResBehav

**Resident Service Level Report (SeniorResSerLev)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorResidentServiceLevelReport.txt

Details

*YSR Template Names*

rs\_SeniorResidentServiceLevelReport.xls

Details

**Resident Service Plan (EHRSP)***Top-Level Select Script Files*

rs\_SeniorEHRXServicePlanYSR.txt

*YSR Script Files*

rs\_SeniorEHRXServicePlan.txt

EHRSP

rs\_SeniorEHRXServicePlan\_SignOff.txt

SignOff

*YSR Template Names*

rs\_SeniorEHRXServicePlan.xlsx

EHRSP

rs\_SeniorEHRXServicePlan\_signOff.xlsx

SignOff

**Resident Service Plan (EHRSPLink)***Top-Level Select Script Files*

rs\_SeniorEHRXServicePlanYSR.txt

*YSR Script Files*

rs\_SeniorEHRXServicePlan.txt

EHRSP

rs\_SeniorEHRXServicePlan\_SignOff.txt

SignOff

*YSR Template Names*

rs_SeniorEHRXServicePlan.xlsx	EHRSP
rs_SeniorEHRXServicePlan_signOff.xlsx	SignOff

**Resident Wound (EHRXResWound)***Top-Level Select Script Files*

rs\_SeniorEHRxResidentWound.txt

*YSR Script Files*

rs\_SeniorEHRxResidentWound.txt

SWound

*YSR Template Names*

rs\_SeniorEHRxResidentWound.xlsx

SWound

**Resident Wound Latest Unhealed Entry (EHRXResWoundLUE)***Top-Level Select Script Files*

rs\_SeniorEHRxResidentWoundLatest.txt

*YSR Script Files*

rs\_SeniorEHRxResidentWoundLatest.txt

WoundLUE

*YSR Template Names*

rs\_SeniorEHRxResidentWoundLatest.xlsx

WoundLUE

**SBAR Communication Form Report (EHRSBAR)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRXSBARCommunicationForm.txt

SBAR

*YSR Template Names*

rs\_SeniorEHRXSBARCommunicationForm.xlsx

SBAR

**Senior Aging by Accounting Period (SeniorAging)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorAnalytic.txt

Chr

rs\_SeniorAnalytic.txt

Com

rs\_SeniorAnalytic.txt

Pay

rs\_SeniorAnalytic.txt

Res

*YSR Template Names*

SeniorAgingChargeCode.xlsx

Chr

SeniorAging.xlsx

Com

SeniorAgingPayer.xlsx

Pay

SeniorAgingResident.xlsx

Res

**Senior Conversion Ratios (ConversionRatios)**

*Top-Level Select Script Files**YSR Script Files*

rs\_SeniorConversionRatioYSR.txt

ConRto

*YSR Template Names*

rs\_SeniorConversionRatio.xlsx

ConRto

**Senior Conversion Report (SeniorConv)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorAnalytic.txt

Conv

rs\_SeniorAnalytic.txt

SCon

*YSR Template Names*

SeniorConversionCommunity.xlsx

Conv

SeniorConversionSource.xlsx

SCon

**Senior Lead Source (SeniorLedSou)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorAnalytic.txt

LedSou

*YSR Template Names*

SeniorLeadSource.xlsx

LedSou

**Senior Rent Roll (SeniorRentRoll)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorAnalytic.txt

RentRoll

rs\_SeniorAnalytic.txt

RRUnit

*YSR Template Names*

SeniorRentRoll.xlsx

RentRoll

SeniorRentRollUnit.xlsx

RRUnit

**Senior Resident Days (SeniorResDays)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorAnalytic.txt

ResDays

*YSR Template Names*

SeniorResidentDays.xlsx

ResDays

**Senior Trend - Activity (SeniorTrendAct)***Top-Level Select Script Files*

<i>YSR Script Files</i>		
rs_SeniorAnalytic.txt		TrdActMt
rs_SeniorAnalytic.txt		
TrdActWk		
<i>YSR Template Names</i>		
SeniorTrendActivityMonthly.xlsx		TrdActMt
SeniorTrendActivityWeekly.xlsx		
TrdActWk		
<b>Senior Trend Occupancy (SeniorTrendOcc)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorAnalytic.txt		TrOccM
rs_SeniorAnalytic.txt		TrOccW
<i>YSR Template Names</i>		
SeniorTrendOccupancyMonthly.xlsx		TrOccM
SeniorTrendOccupancyWeekly.xlsx		TrOccW
<b>Senior Unit Availability (SeniorUnitAvai)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorAnalytic.txt		UnitAvai
<i>YSR Template Names</i>		
SeniorUnitAvailability.xlsx		UnitAvai
<b>Senior Weekly Conversion Ratio (SeniorWeekConv)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorAnalytic.txt		WkConv
<i>YSR Template Names</i>		
SeniorWeeklyConversion.xlsx		WkConv
<b>Skin Check Report (EHRSkinCheck)</b>		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_SeniorEHRXSkinCheckReport.txt		EHRSC
<i>YSR Template Names</i>		
rs_SeniorEHRXSkinCheckReport.xlsx		EHRSC
<b>Submission Report - Consolidated (EHRMDSSubCon)</b>		
<i>Top-Level Select Script Files</i>		

*YSR Script Files*

rs\_SeniorEHRXMDSSubmissionConsolidated.txt  
EHRMDSSC

*YSR Template Names*

rs\_SeniorEHRXMDSSubmissionConsolidated.xlsx  
EHRMDSSC

**Supervisory Assessment Log (SuperViLog)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRxSuperVisorAssessmentLog.txt SuperViL

*YSR Template Names*

rs\_SeniorEHRxSupervisorAssessmentLog.xlsx SuperViL

**Supervisory Assessment Report (SuperVi)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRxSupervisorAssessment.txt SuperVi

*YSR Template Names*

rs\_SeniorEHRxSupervisorAssessment.xlsx SuperVi

**Supervisory Visit Due Assessment (SuperViDue)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRxSupervisoryVisitDue.txt SuperVD

*YSR Template Names*

rs\_SeniorEHRxSupervisoryVisitDue.xlsx SuperVD

**Unduplicated Census (EHRHHUndupCen)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRXUnduplicatedCensus.txt d

*YSR Template Names*

rs\_SeniorEHRXUnduplicatedCensus.xlsx d

**Utilization Summary Hour (EHRHHUtilSummHr)***Top-Level Select Script Files**YSR Script Files*

rs\_SeniorEHRXUtilizationSummaryHour.txt d

*YSR Template Names*

rs_SeniorEHRXUtilizationSummaryHour.xlsx	d
<b>Utilization Summary Visit (EHRHHUtilSummVis)</b>	
<i>Top-Level Select Script Files</i>	
YSR Script Files	
rs_SeniorEHRXUtilizationSummaryVisit.txt	d
<i>YSR Template Names</i>	
rs_SeniorEHRXUtilizationSummaryVisit.xlsx	d
<b>Visit Status (EHRHHVisitStatus)</b>	
<i>Top-Level Select Script Files</i>	
YSR Script Files	
rs_SeniorEHRXVisitStatus.txt	d
<i>YSR Template Names</i>	
rs_SeniorEHRXVisitStatus.xlsx	d

## Single Family Homes

### Single Family Rent Roll (SFHRentRoll)

<i>Top-Level Select Script Files</i>	
YSR Script Files	
YSR Template Names	
select case when '#Summary#='Home' and '#SumryCheck#=0' and '#Opstatus#=1' then 'YSR_SFHHomeNonSummaryWithOperatingStatus.xlsx' when '#Summary#='Home' and '#SumryCheck#=0' and '#Opstatus#=0' then 'YSR_SFHHomeNonSummaryNoOperatingStatus.xlsx' when ('#Summary#='Home' and '#SumryCheck#=1') or ('#Summary#='submarket' and '#SumryCheck#=0') then 'YSR_SFH_HomSummarySubMNonSummary.xlsx' when ('#Summary#='submarket' and '#SumryCheck#=1') OR ('#Summary#='Home' and '#SumryCheck#=1') OR ('#Summary#='Market' and '#SumryCheck#=0')then 'YSR_SFH_SubMSummaryMarketNonSummary.xlsx' when '#Summary#='Market' and	SFHRR

## Tax and Insurance

### TI Commercial Insurance Policies (TICINSPOL)

<i>Top-Level Select Script Files</i>	
rs_TI_CommlInsurancePolicies.txt	
YSR Script Files	
rs_TI_CommlInsurancePolicies.txt	INSUDET

<i>YSR Template Names</i>		
ydoc_TI_CommlnInsurancePolicies.xlsx		INSUDET
<b>TI Insurance Claims (INSCLAIM)</b>		
<i>Top-Level Select Script Files</i>		
rs_TI_InsuranceClaims.txt		
<i>YSR Script Files</i>		
rs_TI_InsuranceClaims.txt		
INSCLAIM		
<i>YSR Template Names</i>		
ydoc_TI_InsuranceClaims.xlsx		
INSCLAIM		
<b>TI Insurance Coverages (TIINSCOV)</b>		
<i>Top-Level Select Script Files</i>		
rs_TI_InsuranceCovgeries.txt		
<i>YSR Script Files</i>		
rs_TI_InsuranceCovgeries.txt		INSUDET
<i>YSR Template Names</i>		
ydoc_TI_InsuranceCovgeries.xlsx		INSUDET
<b>TI Insurance Relationships (TIINSREL)</b>		
<i>Top-Level Select Script Files</i>		
rs_TI_Relationships.txt		
<i>YSR Script Files</i>		
rs_TI_Relationships.txt		INSREL
<i>YSR Template Names</i>		
ydoc_TI_Relationships.xlsx		INSREL
<b>TI Missing Insurance Policies (TIMISINS)</b>		
<i>Top-Level Select Script Files</i>		
rs_TI_MissingInsurance.txt		
<i>YSR Script Files</i>		
rs_TI_MissingInsurance.txt		MISSINS
<i>YSR Template Names</i>		
ydoc_TI_MissingInsurance.xlsx		MISSINS
<b>TI Tax Appeals (TITAXAPP)</b>		
<i>Top-Level Select Script Files</i>		
rs_TI_TaxAppeals.txt		
<i>YSR Script Files</i>		

rs_TI_TaxAppeals.txt	TAXAPP
<i>YSR Template Names</i> ydoc_TI_TaxAppeals.xlsx	TAXAPP
<b>TI Tax Assessments (TIASMVAL)</b>	
<i>Top-Level Select Script Files</i> rs_TI_TaxAssessments.txt	
<i>YSR Script Files</i> rs_TI_TaxAssessments.txt	
TAXASMTD	
<i>YSR Template Names</i> ydoc_TI_TaxAssessments.xlsx	
TAXASMTD	
<b>TI Tax Assessments (TITAXASM)</b>	
<i>Top-Level Select Script Files</i> rs_TI_TaxAssessments.txt	
<i>YSR Script Files</i> rs_TI_TaxAssessments.txt	
TAXASMTD	
rs_TI_TaxParcels.txt	TAXPAR
<i>YSR Template Names</i> ydoc_TI_TaxAssessments.xlsx	
TAXASMTD	
ydoc_TI_TaxParcels.xlsx	TAXPAR
<b>TI Tax Exemptions (TITAXEXE)</b>	
<i>Top-Level Select Script Files</i> rs_TI_TaxExemptions.txt	
<i>YSR Script Files</i> rs_TI_TaxExemptions.txt	
<i>YSR Template Names</i> ydoc_TI_TaxExemptions.xlsx	
TAXEXE	
<b>TI Tax Parcels (TITAXPAR)</b>	
<i>Top-Level Select Script Files</i> rs_TI_TaxParcels.txt	
<i>YSR Script Files</i> rs_TI_TaxParcels.txt	
<i>YSR Template Names</i> ydoc_TI_TaxParcels.xlsx	
TAXPAR	

**TI Tax Relationships (TITAXREL)***Top-Level Select Script Files*

rs\_TI\_TaxRelationships.txt

*YSR Script Files*

rs\_TI\_TaxRelationships.txt

INSREL

*YSR Template Names*

ydoc\_TI\_TaxRelationships.xlsx

INSREL

**UnitSales****UnitSaleCRM Proposal Document (UnitSaleCRMPromo)***Top-Level Select Script Files**YSR Script Files*

rs\_UnitSaleCRM\_Proposal.txt

headyDoc

*YSR Template Names*

yDoc\_UnitSaleCRM.docx

headyDoc

**UnitSaleCRM SPA Document (UnitSaleCRMSPA)***Top-Level Select Script Files**YSR Script Files*

rs\_SPA\_Report.txt

CorpJO

rs\_SPA\_Report.txt

First

rs\_SPA\_Report.txt

IndJO

rs\_SPA\_Report.txt

Last

*YSR Template Names*

yDoc\_SPA\_JointOwner\_Corp.docx

CorpJO

yDoc\_SPA\_First.docx

First

yDoc\_SPA\_JointOwner\_Ind.docx

IndJO

yDoc\_SPA\_Last.docx

Last

**UnitSalesCRM PDC (UnitSalesCRMPDC)***Top-Level Select Script Files**YSR Script Files*

rs\_YSR\_UnitSalesCRM\_PDC.txt

P1

*YSR Template Names*

YSR\_UnitSalesCRM\_PDC.xlsx

P1

## Unknown

### Booking Schedule (CML\_BS)

*Top-Level Select Script Files*

rs\_sql\_BookingFunction.txt

*YSR Script Files*

rs\_sql\_BookingFunction.txt

std01

*YSR Template Names*

Ydoc\_Booking\_Schedule.docx

std01

### CML Amendment Abstract (CML\_AA)

*Top-Level Select Script Files*

CMLAmendmentAbstractMainQuery.txt

*YSR Script Files*

rs\_sql\_CMLAmendmentAbstractScript.txt

aa02

*YSR Template Names*

YSR\_CMLAmendmentAbstractTemplate.xlsx

aa02

### CML Proposal Abstract (CMLProposal\_Abst)

*Top-Level Select Script Files*

CMLProposalAbstractMainQuery.txt

*YSR Script Files*

rs\_sql\_CMLProposalAbstractScript.txt

aa02

*YSR Template Names*

YSR\_CMLProposalAbstractTemplate.xlsx

aa02

### Description of MDSSchedulerMon (MDSSchedulerMon)

*Top-Level Select Script Files*

MDSSchedulerMonthView.txt

*YSR Script Files*

MDSSchedulerMonthView.txt

rpt

*YSR Template Names*

MDSSchedulerMonthView.xlsx

rpt

### Description of MDSSchedulerRep (MDSSchedulerRep)

*Top-Level Select Script Files*

MDSSchedulerMonthReportView.txt

*YSR Script Files*

MDSSchedulerMonthReportView.txt

rpt

*YSR Template Names*

MDSSchedulerMonthReportView.xlsx	rpt
<b>Lease Abstract Report (CML_LAR)</b>	
<i>Top-Level Select Script Files</i>	
LeaseAbstractMainQuery.txt	
<i>YSR Script Files</i>	
rs_sql_LeaseAbstractScript.txt	rc01
<i>YSR Template Names</i>	
YSR_LeaseAbstractTemplate.docx	rc01
<b>MDSCA CI Report (MDSCA_CI)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
MDSCA_Report_CI.txt	
MDSCA_CI	
<i>YSR Template Names</i>	
MDSCA_Report_CI.xlsx	
MDSCA_CI	
<b>MDSCA SP Report (MDSCA_SP)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
MDSCA_Report_SP.txt	
MDSCA_SP	
<i>YSR Template Names</i>	
MDSCA_Report_SP.xlsx	
MDSCA_SP	
<b>MDSCA UP Report (MDSCA_UP)</b>	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
MDSCA_Report_UP.txt	
MDSCA_UP	
<i>YSR Template Names</i>	
MDSCA_Report_UP.xlsx	
MDSCA_UP	
<b>Specialty Amendment Abstract (Spec_AA)</b>	
<i>Top-Level Select Script Files</i>	
SpecialtyAmendmentAbstractMainQuery.txt	
<i>YSR Script Files</i>	
rs_sql_SpecialtyAmendmentAbstractScript.txt	aa02

<i>YSR Template Names</i>		
YSR_SpecialtyAmendmentAbstractTemplate.xlsx		aa02
<b>Specialty Proposal Abstract (Spec_Prop)</b>		
<i>Top-Level Select Script Files</i>		
SpecialtyProposalAbstractMainQuery.txt		
<i>YSR Script Files</i>		
rs_sql_SpecialtyProposalAbstractScript.txt		aa02
<i>YSR Template Names</i>		
YSR_SpecialtyProposalAbstractTemplate.xlsx		aa02
<b>Standard Tax Invoice (CML_TaxInv)</b>		
<i>Top-Level Select Script Files</i>		
rs_TaxInvoice_DetailScript.txt		
<i>YSR Script Files</i>		
rs_TaxInvoice_DetailScript.txt		Invoice
<i>YSR Template Names</i>		
yDoc_TaxInvoice.docx		Invoice

## Utility Billing

### Monthly Billing Recap Report-Summary (UB\_MBR)

<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_UtilityBilling_MonthlyBillingRecap_Summary.txt		UB_MBR
<i>YSR Template Names</i>		
rs_UtilityBilling_MonthlyBillingRecap_Summary.xlsx		UB_MBR

### Utility Billing All Meters (UBMeterSetupAll)

<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
rs_YSR_Utility_Billing_Meter_Setup_All_Report.txt		
UBMeterS		
<i>YSR Template Names</i>		
YSR_Utility_Billing_Meter_Setup_All_Report.xlsx		
UBMeterS		

### Utility Billing Invoice (UB\_Invoice)

<i>Top-Level Select Script Files</i>		
rs_YSR_InternationalMEA_UB_Invoice_Header.TXT		

*YSR Script Files*

rs\_YSR\_InternationalMEAA\_UB\_Invoice\_Detail.TXT rpt

*YSR Template Names*

YSR\_InternationalMEAA\_UB\_Invoice.docx rpt

**Utility Billing Missing Meters (UBMeterSetupMis)***Top-Level Select Script Files**YSR Script Files*

rs\_YSR\_Utility\_Billing\_Meter\_Setup\_Missing\_Report.txt

UBMeterS

*YSR Template Names*

YSR\_Utility\_Billing\_Meter\_Setup\_Missing\_Report.xlsx

UBMeterS

**Utility Billing Shared Meters (UBMeterSetupShar)***Top-Level Select Script Files**YSR Script Files*

rs\_YSR\_Utility\_Billing\_Meter\_Setup\_Shared\_Report.txt

UBMeterS

*YSR Template Names*

YSR\_Utility\_Billing\_Meter\_Setup\_Shared\_Report.xlsx

UBMeterS

**Utility Billing Trend Analysis (UBTrendAnalysis)***Top-Level Select Script Files**YSR Script Files*

rs\_YSR\_Utility\_Billing\_Trend\_Analysis\_Report.txt

TrendAna

*YSR Template Names*

YSR\_Utility\_Billing\_Trend\_Analysis\_Report.xlsx

TrendAna

**Wait****Waiting List Family History (WLFamilyHis)***Top-Level Select Script Files*

rs\_WL\_YSR\_Waiting\_List\_FamHist.txt

*YSR Script Files*

rs\_WL\_YSR\_Waiting\_List\_FamHist.txt WL

*YSR Template Names*

```
SELECT CASE '#RptType#' WHEN 'Application Review' THEN
'rx_WL_YSR_Waiting_List_FamHist.xlsx' ELSE 'rx_WL_YSR_Waiting_List_FamHist_Position.xlsx'
```

WL

## YES

### Annual Review Report (ARR)

*Top-Level Select Script Files*

*YSR Script Files*

rs_UB_UtilityBillingAnnualReview.txt	PropDet
rs_UB_UtilityBillingAnnualReviewSummary.txt	propSum

*YSR Template Names*

rs_UB_UtilityBillingAnnualReview.xlsx	PropDet
rs_UB_UtilityBillingAnnualReviewSummary.xlsx	propSum

### Utility Billing Corporate Report (Corporat)

*Top-Level Select Script Files*

*YSR Script Files*

ss_UtilityBilling_CorporateReport.txt	crprpt
---------------------------------------	--------

*YSR Template Names*

ss_UtilityBilling_CorporateReport.xlsx	crprpt
--	--------

### YES Revenue KPI (YESKPIRV)

*Top-Level Select Script Files*

*YSR Script Files*

FinancialAnalytic.txt	KPIRPT
-----------------------	--------

*YSR Template Names*

FinRpt.xlsx	KPIRPT
-------------	--------

## YSR Admin

### YSR - Generate Report History (YSRHISTORY)

*Top-Level Select Script Files*

*YSR Script Files*

rs_YSR_GenerateHistory.txt	YSRHist
----------------------------	---------

*YSR Template Names*

select case '#ReportType#' when 'Summary' then 'rs_YSR_GenerateHistory_Summary.xlsx' when 'Detail' then 'rs_YSR_GenerateHistory_Detail.xlsx' when 'Drill' then 'rs_YSR_GenerateHistory_Drill.xlsx' when 'DrillTotal' then 'rs_YSR_GenerateHistory_Drill_Total.xlsx' else 'rs_YSR_GenerateHistory.xlsx' end Template	YSRHist
--	---------

## APPENDIX C

# YSR Reports with Voyager Analytics Data Sources

This section lists the standard YSR reports that use Voyager analytics as data sources. The list is organized by data source.

### Affordable Analytics DataTable

- Affordable GPR Report (AffGPR)
- Affordable Rent Roll with Lease Charges (AffRntRollLsChgs)

### Affordable Analytics SQL

- Affordable Rent Roll (AffRentRoll)
- Affordable Unit Vacancy (AffUnitVacancy)

### AP Analytics

- Owner Monthly Report (OwnerPac)
- Owner Monthly Report - 1 Tax (Ownr1Tax)
- Owner Monthly Report - 2 Taxes (Ownr2Tax)
- Property Report Packet (PropPack)
- Property Report Packet - 1 Tax (Prop1Tax)
- Property Report Packet - 2 Taxes (Prop2Tax)

### AR Analytics

- Aging Detail Report (AginDet)
- AR Aging Summary Report (AgingSum)
- Owner Monthly Report - 1 Tax (Ownr1Tax)
- Owner Monthly Report - 2 Taxes (Ownr2Tax)
- Property Report Packet - 1 Tax (Prop1Tax)
- Property Report Packet - 2 Taxes (Prop2Tax)

### AR Analytics Domestic

- Owner Monthly Report (OwnerPac)
- Owner Monthly Report - 1 Tax (Ownr1Tax)

Owner Monthly Report - 2 Taxes (Ownr2Tax)  
 Property Report Packet (PropPack)  
 Property Report Packet - 1 Tax (Prop1Tax)  
 Property Report Packet - 2 Taxes (Prop2Tax)

## Cash Flow PTD SQL

Owner Monthly Report (OwnerPac)  
 Owner Monthly Report - 1 Tax (Ownr1Tax)  
 Owner Monthly Report - 2 Taxes (Ownr2Tax)  
 Property Report Packet (PropPack)  
 Property Report Packet - 1 Tax (Prop1Tax)  
 Property Report Packet - 2 Taxes (Prop2Tax)

## Cash Flow YTD SQL

Owner Monthly Report (OwnerPac)  
 Owner Monthly Report - 1 Tax (Ownr1Tax)  
 Owner Monthly Report - 2 Taxes (Ownr2Tax)  
 Property Report Packet (PropPack)  
 Property Report Packet - 1 Tax (Prop1Tax)  
 Property Report Packet - 2 Taxes (Prop2Tax)

## Commercial Analytics

Commercial Rent Roll (RentRoll)  
 Contact Directory (Contact)  
 Critical Dates (CriDates)  
 Customer Directory (CustDir)  
 Customer Top X (CustTopX)  
 Deal Directory (DealDir)  
 Lease Activity History (LeActHis)  
 Option Expiration (OptExp)  
 Owner Monthly Report (OwnerPac)  
 Owner Monthly Report - 1 Tax (Ownr1Tax)  
 Owner Monthly Report - 2 Taxes (Ownr2Tax)  
 Portfolio Summary (PortSmry)  
 Property Report Packet (PropPack)  
 Property Report Packet - 1 Tax (Prop1Tax)  
 Property Report Packet - 2 Taxes (Prop2Tax)  
 Property Summary (PropSmry)  
 Straight-Line By Lease Summary (STLSumm)  
 Straight-Line JE Detail (STLJEDet)

Straight-Line JE Register (STLJEReg)  
Unpaid Charges (UnpChg)

## Condo Accounts Receivable Aging

Condo Accounts Receivable Aging (CondoAR)  
RentCafe Condo Board Member Reports (RentCafeCondo)  
RentCafeCondo Aged Receivable Report (RentCafeCondoAR)

## Condo Correspondence Invoice

Correspondence Condo Invoice (CondoInvoice)

## Condo Correspondence Ledger

Correspondence Condo Ledger (cndlledger)

## Condo Correspondence Letter

Assoc Arrears Notice Removal From PAP (AssocNoticeRemov)  
Assoc Arrears Reminder Letter (AssocArrears)  
Assoc Arrears Reminder NSF Letter (AssocArrearsNSF)  
Assoc Arrears Reminder3 Caveat Letter (AssocCaveat)  
Assoc Error On Cheque Letter (AssocErrorChk)  
Assoc Foreclosure Notice Letter (AssocForeclosure)  
Assoc Welcome Condo Letter (AssocWelcomeC)  
Assoc Welcome HOA Letter (AssocWelcome)  
Correspondence Condo Letter (CndLetter)

## Condo Correspondence Purchase

Correspondence Condo Purchase (CondoPurchase)

## Condo Owner Ledger Analytic

Condo Analytic Owner Ledger with Aging (Analyticledger)  
Condo Analytic Owner Ledger without Aging (AnalyticLedger1)

## Correspondence (International)

Correspondence International Invoice (CMInv)

## CRM Analytics

CRM Analytics - Consolidated Report (CRMAnaly)

## Custom Financials

Report Packet (RprtPkt)

## Custom IM Data

Custom IM Investment Attribute Report (custIM\_invsatt)  
Custom IM Investment Holding Report (CustIM\_Invstmtnt)

Custom IM Investor Attribute Report (cusIM\_invattr)  
 Custom IM Investor Holding Report (CustIM\_invstr)

## Domestic Transaction Registers

Owner Monthly Report (OwnerPac)  
 Owner Monthly Report - 1 Tax (Ownr1Tax)  
 Owner Monthly Report - 2 Taxes (Ownr2Tax)  
 Property Report Packet (PropPack)  
 Property Report Packet - 1 Tax (Prop1Tax)  
 Property Report Packet - 2 Taxes (Prop2Tax)

## Financial Analytics

Balance Sheet (BalanceS)  
 Budget Comparison (BudgetCo)  
 Income Statement (IncomeSt)  
 Owner Monthly Report (OwnerPac)  
 Owner Monthly Report - 1 Tax (Ownr1Tax)  
 Owner Monthly Report - 2 Taxes (Ownr2Tax)  
 Property Report Packet (PropPack)  
 Property Report Packet - 1 Tax (Prop1Tax)  
 Property Report Packet - 2 Taxes (Prop2Tax)  
 Trial Balance (TrialBal)

## Generic Performance Analytics

Generic Performance Analytics (PerfAnalytic)

## IM Attribute Data

ILPA Fund Report (FundRpt)

## IM Capital Roll Forward Analytics

Capital Balance Report Investment View (cbal\_inv)  
 Capital Balance Report Investor View (cap\_bal)  
 Transaction Detail Report Investment View (tran\_invest)  
 Transaction Detail Report Investor View (Tran\_det)

## IM Custom Financial Data

ILPA Fund Report (FundRpt)

## IM Data

ILPA Fund Report (FundRpt)

## Inspection Analytics DataTable

Inspection Summary Report (INSPSummary)

## Inspection Analytics SQL

- Inspection Details Report (INSPDetail)
- Inspection Duration Report (INSPDuration)
- Inspection Summary Report (INSPSummary)
- Inspection Unassigned Directory Report (INSPUnassigned)
- Inspection Uncompleted Date Review Report (INSPUncompleted)

## International Residential Analytics

- International Residential Lease Expiration (IntResLeaseExp)
- International Residential Market Rent Schedule (IntResMktRentSch)
- International Residential Potential Rent (IntResPotRent)
- International Residential Unit Availability (IntResUnitAva)

## Intl Transaction Registers

- Charge Register (ChargReg)
- Invoice Register (InvoiceR)
- Owner Monthly Report - 1 Tax (Ownr1Tax)
- Owner Monthly Report - 2 Taxes (Ownr2Tax)
- Payable Register (PayableR)
- Property Report Packet - 1 Tax (Prop1Tax)
- Property Report Packet - 2 Taxes (Prop2Tax)
- Receipt Register (Receipt)

## Residential Analytics

- Residential Resident Activity (ResActivity)
- Residential Traffic Detail (ResTrafficDetail)
- Residential Unit Availability (ResUnitAvlbl)

## Senior Community Analytics

- Senior Aging by Accounting Period (SeniorAging)
- Senior Conversion Report (SeniorConv)
- Senior Lead Source (SeniorLedSou)
- Senior Rent Roll (SeniorRentRoll)
- Senior Resident Days (SeniorResDays)
- Senior Trend - Activity (SeniorTrendAct)
- Senior Trend Occupancy (SeniorTrendOcc)
- Senior Unit Availability (SeniorUnitAvai)
- Senior Weekly Conversion Ratio (SeniorWeekConv)

## SFH Analytics Data Table

- Single Family Rent Roll (SFHRentRoll)

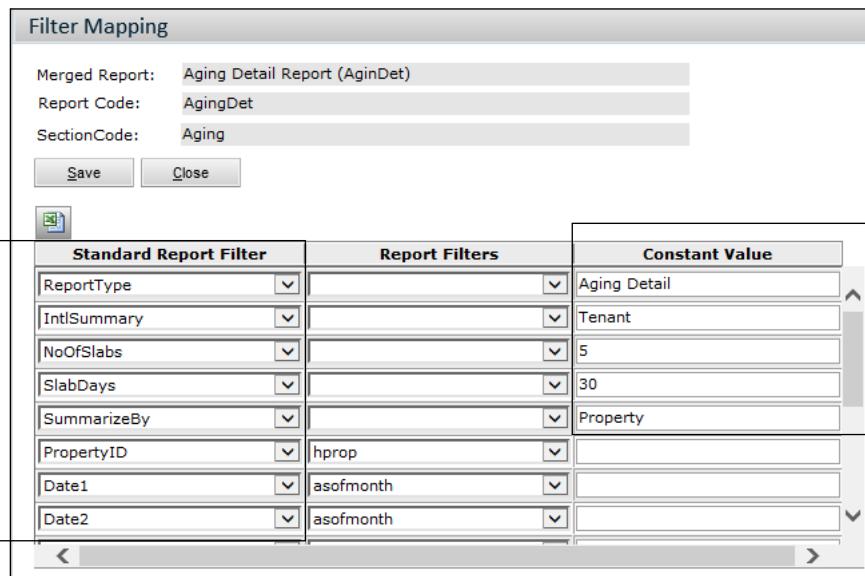
## Tenancy Schedule

Tenancy Schedule (Tenancy)

## APPENDIX D

# Constant Values for Analytics Data Source Filter Mapping

This section provides an exhaustive list of the values that have been used, to date, as constant values for Voyager analytics data sources in YSR. For example, the Aging Detail Report, one of the standard, out-of-the-box YSR reports available with the Financial Analytics Plug-in, uses A/R Analytics to retrieve data. The report assigns the constant value **Aging Detail** to the **ReportType** filter element native to A/R Analytics.



The screenshot shows the 'Filter Mapping' dialog box. At the top, there are three input fields: 'Merged Report' (Aging Detail Report (AginDet)), 'Report Code' (AgingDet), and 'SectionCode' (Aging). Below these are 'Save' and 'Close' buttons. The main area is a grid titled 'Filter Mapping' with columns: 'Standard Report Filter', 'Report Filters', and 'Constant Value'. A vertical line labeled 'A/R Analytics filter elements' points to the first column. The grid contains the following data:

Standard Report Filter	Report Filters	Constant Value
ReportType		Aging Detail
IntlSummary		Tenant
NoOfSlabs		5
SlabDays		30
SummarizeBy		Property
PropertyID	hprop	
Date1	asofmonth	
Date2	asofmonth	

The data listed in this appendix comes from the entire suite of YSR reports authored for inclusion in Voyager Plug-ins, as well as those created historically in response to YSR Custom Programming Requests by Yardi Systems.

The appendix includes all values that appear as constant values as well as those passed down from sets of list values defined for display in the custom filter fields that users complete at run-time. Values that are database-specific (such as the code for a Custom Financial Analytics report template, or the template code for a custom Investment Management report) are excluded from this appendix. Date-referencing fields and ID-specific values are excluded as well.

Where a report designer authored a caret-delimited list for the **Constant Value** field, those lists have been separated into their component values, with each list item appearing on a new row.

When you map filters, you can pass in the values listed below as isolated constants. This is effectively equivalent to hard-coding one aspect of your report design. Alternatively, you can define a custom filter field for your YSR report filter that enables users to select from a caret-delimited list of values at run time. Then you map your custom filter field to the corresponding filter element of the Analytics data source. In that manner, YSR can map and pass the user's filter selection to the underlying Analytics data source.

In some cases you can also assign multiple caret-delimited string values as constant values. For example, you can assign a constant value such as **Accrual^Allocation** to the **BookCodeList** filter element of a Custom Financial Analytics report section. The Custom Financial Analytics data source then aggregates financial activity across multiple G/L books.

In at least a single case, the **PerfField** (Performance field) filter element of IM Performance Data, it has been established that if you pass in a caret-delimited list of Performance field names, those field names are exposed in the resolution of the data source and can therefore be included in your report template design as smart markers.

**Filter Mapping**

Merged Report:	Investor Investment Report (IMReport)
Report Code:	Report
SectionCode:	fundperf
<b>Save</b>	<b>Close</b>

**Standard Report Filter**

PerfTable	ASCAP_AssetInfo
PerfField	dCashBalance^dForecastGain^dFo
ReportStyle	investmentview
PropertyCode	Investor
FromMMYY	FromDate
ToMMYY	ToDate

**Report Filters**


**Constant Value**

Cash Balance	&=fundperf.dCashBalance
Forecast Gain	&=fundperf.dForecastGain
Forecast IRR	&=fundperf.dForecastIRR
Share Price	&=fundperf.dSharePrice
Market Value	&=fundperf.A_BEGINNINGMARKETVALUE

**Fund Performance Data :**

Cash Balance	&=fundperf.dCashBalance
Forecast Gain	&=fundperf.dForecastGain
Forecast IRR	&=fundperf.dForecastIRR
Share Price	&=fundperf.dSharePrice
Market Value	&=fundperf.A_BEGINNINGMARKETVALUE

## Affordable Analytics DataTable

ReportType	Affordable GPR Report
ReportType	Affordable Rent Roll with Lease Charges
ShowAdminUnits	No
ShowAdminUnits	Yes
SortBy	SortByUnit
SummaryType	All
SummaryType	Charge Code
SummaryType	Group
SummaryType	Property
SummaryType	Unit

## Affordable Analytics SQL

ReportType	Affordable Rent Roll
ReportType	Affordable Unit Vacancy Report
SummaryType	Property
SummaryType	Unit
SummaryType	Unit Type

## AP Analytics

BankCode	vpg05
ProgramType	0
ProgramType	YES
ReportType	Aging
ReportType	Expense Distribution
ReportType	Payment Register
ReportType	Vendor Ledger
ShowDetail	YES
ShowVAT	YES

## AR Analytics

ChargeCode	SELECT " -- #hprop#
ChargeCode	SELECT " -- #hproperty#
ChargeCode	SELECT " --##
ChargeCode	SELECT " --'##'
ChargeCode	SELECT ct.hmy from chargtyp ct inner join lookup l on l.sdesc = ct.scode where l.slistname = N'reschargecode' /* # # */
ChargeCode	SELECT dbo.Select_CommaSeparatedChgCodes() /* ## */
ChargeCode	SELECT scode from chargtyp
ChargeCode	SELECT sDesc from Lookup where sListName = 'CommChargeCode' /* " */
IntlSummary	All
IntlSummary	Customer

IntlSummary	Employee
IntlSummary	Owner
IntlSummary	Tenant
IntlSummary	Vendor
LeaseStatus	Applicant
LeaseStatus	Canceled
LeaseStatus	Current
LeaseStatus	Denied
LeaseStatus	Eviction
LeaseStatus	Future
LeaseStatus	Notice
LeaseStatus	Past
LeaseStatus	SELECT status TextField,istatus ValueField from CommTenStatus
LeaseStatus	SELECT DISTINCT ts.istatus [ValueField],ts.status
	[CodeField],ts.status [TextField] from commtenstatus ts where 1=1
LeaseStatus	Vacant
LeaseStatus	Wait List
MinAmount	1
ModuleType	International
NoOfSlabs	1
NoOfSlabs	2
NoOfSlabs	3
NoOfSlabs	4
NoOfSlabs	5
NoOfSlabs	6
OwnerId	"
OwnerId	0
OwnerId	SELECT " -- #hprop#
OwnerId	SELECT " -- #hproperty#
OwnerId	SELECT hmyperson from owner
OwnerId	SELECT hmyperson from owner where 1=1 -- ##
ReportType	Aging Detail
ReportType	Aging Summary
ReportType	Receivable Detail
ReportType	Receivable Summary
RunToday	1
ShowNonTenants	Yes
SlabDays	30
SlabDays	300
SlabDays	45
SlabDays	60

SummarizeBy	Lease By Charge Code
SummarizeBy	Lease/Payee
SummarizeBy	Property
SummarizeBy	Property by Charge Code
SummarizeBy	Tenant

## AR Analytics Domestic

NoOfSlabs	3
ReportType	Aging Detail
ReportType	Aging Summary
ShowNonTenants	Yes
SlabDays	30
SummarizeBy	Tenant

## Cash Flow PTD SQL

FinType	Cash Flow
SuppressZero	1

## Cash Flow YTD SQL

FinType	Cash Flow
SuppressZero	1

## Commercial Analytics

AttributeCode	SELECT sname TextField ,sname ValueField from attributename where ifiletype=3 order by hmy
AttributeValue	SELECT sName TextField, Hmy ValueField FROM AttributeName WHERE iFileType = 3
ContactRole	SELECT 0 ValueField, " TextField UNION ALL select hMy ValueField, sDesc TextField from Role where 1=1 and sObjectType = 'Commercial Lease' order by 2
IsDetail	1
IsDetail	No
IsDetail	True
IsDetail	Yes
IsMarketRent	Yes
LeaseStatus	Current
LeaseStatus	Future
LeaseStatus	Past
MainQuery	No
MainQuery	Yes
PeriodType	Annually
PeriodType	Monthly
PeriodType	Quarterly
ReportType	ContactDirectory

ReportType	CriticalDates
ReportType	CustomerDirectory
ReportType	CustomerTopx
ReportType	DealDirectory
ReportType	LeasingActivityHistory
ReportType	OptionExpiration
ReportType	PortfolioSummary
ReportType	PropertySummary
ReportType	Rent Projection
ReportType	RentRoll
ReportType	RR34446
ReportType	StraightLineByLeaseSummary
ReportType	StraightlineRentsJEDetail
ReportType	StraightlineRentsJERegister
ReportType	UnpaidCharges
SummaryType	Attribute
SummaryType	Brand
SummaryType	Brand Owner
SummaryType	Customer
SummaryType	Property
SummaryType	Type
TopXFact	Area
TopXFact	Rent

## Condo Accounts Receivable Aging

ExcludeUnits	No
ExcludeUnits	Yes
GroupBy	MasterUnit
GroupBy	Owner
GroupBy	Unit
IsChargeCode	No
IsChargeCode	Yes
PaymentType	Any
PaymentType	Cash Only
PaymentType	Do not accept
PaymentType	On Hold
Status	Current
Status	Past

## Condo Correspondence Invoice

PayableMethod	ACH/EFT
PayableMethod	Check
Status	Current

Status	Past
--------	------

## Condo Correspondence Ledger

Status	Current
--------	---------

Status	Past
--------	------

## Condo Correspondence Letter

IncludeOwnerEmail	All
-------------------	-----

IncludeOwnerEmail	With Email
-------------------	------------

IncludeOwnerEmail	Without Email
-------------------	---------------

OrderBy	Owner
---------	-------

OrderBy	Unit
---------	------

SingleRow	Owner
-----------	-------

SingleRow	Unit
-----------	------

Status	Current
--------	---------

Status	Past
--------	------

## Condo Correspondence Purchase

SortBy	Date
--------	------

SortBy	Stage
--------	-------

Stage	(0) Applicant
-------	---------------

Stage	(1) Owner
-------	-----------

Stage	(2) Units
-------	-----------

Stage	(3) Charges
-------	-------------

Stage	(4) Preview
-------	-------------

## Condo Owner Ledger Analytic

OwnerStatus	Current
-------------	---------

OwnerStatus	Future
-------------	--------

OwnerStatus	Past
-------------	------

ReportType	Owner Ledger
------------	--------------

ReportType	Owner Ledger With Aging
------------	-------------------------

SummaryType	Owner
-------------	-------

SummaryType	Unit
-------------	------

## CORRESP\_Maint\_FixedAsset

ReportType	Depreciation Detail
------------	---------------------

ReportType	Depreciation Expense
------------	----------------------

ReportType	Depreciation Summary
------------	----------------------

ReportType	Fixed Asset Directory
------------	-----------------------

ReportType	Period Report
------------	---------------

ReportType	Transaction Detail
------------	--------------------

ReportType	Warranty Detail
------------	-----------------

SummaryType	Accumulated Depr Acct
-------------	-----------------------

SummaryType	Asset Group
-------------	-------------

SummaryType	Asset Type
SummaryType	Capital Acct
SummaryType	Depreciation Exp Acct
SummaryType	Depreciation Group
SummaryType	Property
SummaryType	Property-Unit
SummaryType	Status
SummaryType	Supplier
SummaryType	Warranty Vendor

## CORRESP\_Maint\_FixedAsset\_DataTable

ReportType	Depreciation Detail
ReportType	Depreciation Expense
ReportType	Depreciation Summary
ReportType	Fixed Asset Directory
ReportType	Period Report
ReportType	Transaction Detail
ReportType	Warranty Detail
SummaryType	Accumulated Depr Acct
SummaryType	Asset Group
SummaryType	Asset Type
SummaryType	Capital Acct
SummaryType	Depreciation Exp Acct
SummaryType	Depreciation Group
SummaryType	Property
SummaryType	Property-Unit
SummaryType	Status
SummaryType	Supplier
SummaryType	Warranty Vendor

## CORRESP\_Maint\_Inventory

CalculateValue	Inventory Billing Price
CalculateValue	Inventory Cost Price
CalculateValue	Item Type Billing Price
CalculateValue	Item Type Cost Price
ReportType	Inventory Activity
ReportType	Inventory Analysis
ReportType	Inventory Closing Stock
ReportType	Inventory Detail
ReportType	Inventory Location
ReportType	Inventory Pricing Exception
ReportType	Inventory Reconciliation
ReportType	Inventory Reorder

ReportType	Inventory Supplier
ReportType	Inventory Transfer
ReportType	Physical Inventory
SortBy	Aisle/bin
SortBy	Closing Value
SortBy	Date Last Received
SortBy	Date Last Used
SortBy	GI Property
SortBy	Inventory Billing Price
SortBy	Inventory Cost Price
SortBy	Inventory Location
SortBy	Item Type
SortBy	Item Type Billing Price
SortBy	Item Type Category
SortBy	Item Type Cost Price
SortBy	Item Type Description
SortBy	Location Type
SortBy	Operation Date
SortBy	Operation Type
SortBy	Person Responsible
SortBy	Qty Difference
SortBy	Quantity Available
SortBy	Quantity Pending
SortBy	Qunty On Hand
SortBy	Total Used
SortBy	Total Value
SortBy	Value
SortBy	Value Difference
SortBy	Variance
SortBy	Vendor
SummaryType	Aisle Bin
SummaryType	Break Down Item Type
SummaryType	Category
SummaryType	GI Property
SummaryType	Inventory Location
SummaryType	Item Description
SummaryType	Item Type
SummaryType	Location Type
SummaryType	Operation Date
SummaryType	Operation Type
SummaryType	Person Responsible

SummaryType	Property
SummaryType	Transfer No
SummaryType	Vendor

## CORRESP\_Maint\_Inventory\_DataTable

CalculateValue	Inventory Billing Price
CalculateValue	Inventory Cost Price
CalculateValue	Item Type Billing Price
CalculateValue	Item Type Cost Price
ReportType	Inventory Activity
ReportType	Inventory Analysis
ReportType	Inventory Closing Stock
ReportType	Inventory Detail
ReportType	Inventory Location
ReportType	Inventory Pricing Exception
ReportType	Inventory Reconciliation
ReportType	Inventory Reorder
ReportType	Inventory Supplier
ReportType	Inventory Transfer
ReportType	Physical Inventory
SortBy	Aisle/bin
SortBy	Closing Value
SortBy	Date Last Received
SortBy	Date Last Used
SortBy	GI Property
SortBy	Inventory Billing Price
SortBy	Inventory Cost Price
SortBy	Inventory Location
SortBy	Item Type
SortBy	Item Type Billing Price
SortBy	Item Type Category
SortBy	Item Type Cost Price
SortBy	Item Type Description

SortBy	Location Type
SortBy	Operation Date
SortBy	Operation Type
SortBy	Person Responsible
SortBy	Qty Difference
SortBy	Quantity Available
SortBy	Quantity Pending
SortBy	Qunty On Hand
SortBy	Total Used
SortBy	Total Value
SortBy	Value
SortBy	Value Difference
SortBy	Variance
SortBy	Vendor
SummaryType	Aisle Bin
SummaryType	Break Down Item Type
SummaryType	Category
SummaryType	GI Property
SummaryType	Inventory Location
SummaryType	Item Description
SummaryType	Item Type
SummaryType	Location Type
SummaryType	Operation Date
SummaryType	Operation Type
SummaryType	Person Responsible
SummaryType	Property
SummaryType	Transfer No
SummaryType	Vendor

## CORRESP\_Maint\_PlannedMaintenance

ReportType	Asset Cost In Ppm
ReportType	Compliance Task
ReportType	Expiring Service Contract Within 1RP Days
ReportType	Outstanding Task Po
ReportType	Service Contract Coverage
ReportType	Service Contract Exception
ReportType	Task Analysis By Month
ReportType	Task Exception

## CORRESP\_Maint\_WorkOrder

ReportType	Aging
ReportType	Average Per Day
ReportType	Completed Activity

ReportType	Completion Aging
ReportType	Completion Detail
ReportType	Completion Summary
ReportType	Cost Analysis
ReportType	Day
ReportType	Directory
ReportType	Employee Activity
ReportType	Employee Calendar
ReportType	Employee Directory
ReportType	Month
ReportType	Open Work Order
ReportType	Performance Summary
ReportType	Period Activity
ReportType	Period Overview
ReportType	Priority
ReportType	Status
SortBy	Active
SortBy	AvgHoursPerWO
SortBy	CallDate
SortBy	Category
SortBy	CompletedDate
SortBy	EmployeeCode
SortBy	FirstName
SortBy	LastName
SortBy	MaxWOTime
SortBy	Priority
SortBy	Property
SortBy	Status
SortBy	SubCategory
SortBy	TotalActualHours
SortBy	Unit
SortBy	User
SortBy	WOCount
SortBy	WorkOrder
SummaryType	Asset
SummaryType	BillTo
SummaryType	Category
SummaryType	DueDate
SummaryType	Employee
SummaryType	Priority
SummaryType	Property

SummaryType	PropertyAndUnit
SummaryType	RelatedWO
SummaryType	Resolution
SummaryType	Role
SummaryType	Skill
SummaryType	Status
SummaryType	SubCategory
SummaryType	Template
SummaryType	Vendor

## CORRESP\_Maint\_WorkOrder\_DataTable

ReportType	Aging
ReportType	Average Per Day
ReportType	Completed Activity
ReportType	Completion Aging
ReportType	Completion Detail
ReportType	Completion Summary
ReportType	Cost Analysis
ReportType	Day
ReportType	Directory
ReportType	Employee Activity
ReportType	Employee Calendar
ReportType	Employee Directory
ReportType	Month
ReportType	Open Work Order
ReportType	Performance Summary
ReportType	Period Activity
ReportType	Period Overview
ReportType	Priority
ReportType	Status
SortBy	Active
SortBy	AvgHoursPerWO
SortBy	CallDate
SortBy	Category
SortBy	CompletedDate
SortBy	EmployeeCode
SortBy	FirstName
SortBy	LastName
SortBy	MaxWOTime
SortBy	Priority
SortBy	Property
SortBy	Status

SortBy	SubCategory
SortBy	TotalActualHours
SortBy	Unit
SortBy	User
SortBy	WOCount
SortBy	WorkOrder
SummaryType	Asset
SummaryType	BillTo
SummaryType	Category
SummaryType	DueDate
SummaryType	Employee
SummaryType	Priority
SummaryType	Property
SummaryType	PropertyAndUnit
SummaryType	RelatedWO
SummaryType	Resolution
SummaryType	Role
SummaryType	Skill
SummaryType	Status
SummaryType	SubCategory
SummaryType	Template
SummaryType	Vendor

## Correspondence (International)

ChargeType	All
ChargeType	CAM
ChargeType	MISC
ChargeType	OVG
ChargeType	RENT
IsCopiesOnly	SELECT CASE WHEN #MARKCOPY#=0 THEN 'NO' ELSE 'YES' END
IsOriginalOnly	SELECT CASE WHEN #MARKCOPY#=1 THEN 'NO' ELSE 'YES' END
IsReminderTypeInvoiceNumberTrue	
OrderBy	ChargeID
OrderBy	CtrlNo
ReportType	10
ReportType	11
ReportType	12
ReportType	2
ReportType	9
ReportType	SELECT case '#Letter#' when 'Reminder 1' then 10 when 'Reminder 2' then 11 when 'Reminder 3' then 12 end
SectionType	HEADER

## CRM Analytics

DealStages	SELECT " TextField," ValueField UNION select scode + ' - ' + sdesc TextField ,scode + ' - ' + sdesc ValueField from CommDealStatus
ObjectType	1783
ObjectType	508
ObjectType	C542
ObjectType	F542
ObjectType	P542
ReportType	CRMAnalytics
ShowUnassignedLeads	No
ShowUnassignedLeads	Yes
SummaryType	Customer
SummaryType	Leasing Agent
SummaryType	Property

## Custom Financials

AccountCode	aip_is
AccountCode	test_aip
AttributeSortName	Consolidation Status
AttributeSortName	Property or Entity
AttributeSortName	PropertyOrEntity
AttributeSortName	SELECT an.sName ValueField , an.sName TextField from AttributeName an WHERE 1=1 Union SELECT 'Property Or Entity' ValueField , 'Property Or Entity' TextField from AttributeName an WHERE 1=1
AttributeSortName	SELECT distinct an.sName TextField, an.sName ValueField from AttributeName an UNION ALL SELECT 'Property Or Entity','Property Or Entity' where 1=1
AttributeSortValue	Consolidation Status
AttributeSortValue	Property or Entity
AttributeSortValue	PropertyOrEntity
AttributeSortValue	SELECT an.sName ValueField , an.sName TextField from AttributeName an WHERE 1=1 Union SELECT 'Property Or Entity' ValueField , 'Property Or Entity' TextField from AttributeName an WHERE 1=1
AttributeSortValue	SELECT distinct an.sName TextField, an.sName ValueField from AttributeName an UNION ALL SELECT 'Property Or Entity','Property Or Entity' where 1=1
Budget	-1
DecimalDigits	0
DecimalDigits	1
DecimalDigits	2

DecimalDigits	3
DecimalDigits	4
DecimalDigits	5
Detail	1
Detail	No
Detail	True
Detail	Yes
DisplayCode	ShowColumn
FinType	DETAILED INCOME STATEMENT
Grid	1
Grid	FALSE
IncludeNotes	1
IncludeNotes	Y
IsConsolidate	0
IsConsolidate	1
IsConsolidate	-1
IsConsolidate	FALSE
IsConsolidate	No
IsConsolidate	True
IsConsolidate	Yes
ShowTreeSummary	0
ShowTreeSummary	1
SummaryBy1	Property Or Entity
SummaryBy2	Property Or Entity
SuppressHeader	1
SuppressHeader	TRUE
SuppressTotalRows	1
SuppressTotalRows	True
SuppressZero	0
SuppressZero	1
SuppressZero	False
SuppressZero	No
SuppressZero	SELECT Case '#bSupressZero#' When 'Yes' then 'TRUE' Else 'False'
SuppressZero	SELECT case when '#SupressZero#' = 'Yes' then 1 else 0 end
SuppressZero	true
SuppressZero	Yes
TreeLevel	1
TreeLevel	10
TreeLevel	2
TreeLevel	3
TreeLevel	4

TreeLevel	5
TreeLevel	6
TreeLevel	7
TreeLevel	8
TreeLevel	9

## Custom IM Data

AttributeName	SELECT sname TextField, Sname ValueField from AttributeName
DateInUse	Effective Date
DateInUse	Post Month
DateInUse	Tran Date
DecimalDigit	0
DecimalDigit	1
DecimalDigit	10
DecimalDigit	2
DecimalDigit	3
DecimalDigit	4
DecimalDigit	5
DecimalDigit	6
DecimalDigit	7
DecimalDigit	8
DecimalDigit	9
Detail	FALSE
LevelID	SELECT ilevel ValueField ,stype TextField from FUND_LEVELS where 1=1 union all select 0,'Property' where 1=1 union all select - 1,"where 1=1 order by 1
ReportStyle	Investment
ReportStyle	Investor
SuppressZero	1
SuppressZero	True

## Domestic Transaction Registers

Status	All
Status	Paid
Status	Unpaid
TranDetail	1
TranType	Charge
TranType	Cheque
TranType	Journal
TranType	Payable
TranType	Receipt

## FA\_yisiARInterface

ChargeCode	SELECT ct.hmy from chargtyp ct inner join lookup l on l.sdesc = ct.scode where l.slistname = N'reschargecode' /* # # */
ChargeCode	SELECT sDesc from Lookup where sListName = 'CommChargeCode' /* '' */
IntlSummary	All
IntlSummary	Tenant
NoOfSlabs	1
NoOfSlabs	2
NoOfSlabs	4
NoOfSlabs	6
ReportType	Aging Detail
ReportType	Aging Summary
SlabDays	30
SlabDays	300
SlabDays	60
SummarizeBy	Lease/Payee
SummarizeBy	Property
SummarizeBy	Tenant

## Financial Analytics

Budget	GL
DecimalDigits	0
DecimalDigits	1
DecimalDigits	2
DecimalDigits	3
DecimalDigits	4
FinType	BALANCE SHEET
FinType	Budget Comparison
FinType	Cash Flow
FinType	General Ledger
FinType	INCOME STATEMENT
FinType	TRIAL BALANCE
GLAPSummary	Batch
GLARSummary	Batch
GLCashSummary	Batch
GLRentSummary	Batch
Graph	0
Grid	0
Grid	1
IncludeNotes	1
IsConsolidate	0
IsConsolidate	1

IsConsolidate	SELECT Case When '#IsConsolidate#'='Yes' Then 1 Else 0 End
IsConsolidate	TRUE
IsCosolidate	0
SuppressZero	0
SuppressZero	1
SuppressZero	SELECT case when '#SuppressZero#' = 'Yes' then 1 else 0 end
SuppressZero	TRUE
SuppressZero	Yes
TreeLevel	1
TreeLevel	10
TreeLevel	2

## Generic Performance Analytics

TableName	SELECT SDESC TextField, SFIELD ValueField from ASUSRDDF Where 1=1 and SDESC<>'' and STYPE ='TABLE' and STABLENAME like 'ASCAP%'
-----------	---

## IM Attribute Data

AttributeNames	a_Manager_Name
AttributeNames	c_Level
AttributeNames	c_Product
ReportStyle	investmentview
ReportStyle	InvestorView

## IM Capital Roll Forward Analytics

DateInUse	Effective Date
DateInUse	Post Month
DateInUse	Tran Date
DecimalDigits	0
DecimalDigits	1
DecimalDigits	2
DecimalDigits	3
DecimalDigits	4
InvestmentCodes	SELECT Dbo.CBREGIUS_GetProductParams(N'#ProductCode#',N'dynamiccli st',N'',0) -- #PropertyCode#
InvestorCodes	SELECT Dbo.CBREGIUS_GetProductParams(N'#ProductCode#',N'entity',N'', 0) -- #PropertyCode#
InvestorCodes	SELECT dbo.GetInvestorsCPR39464('#Client#','#ToDate#')
LevelCode	Property
LevelID	SELECT illevel ValueField ,stype TextField from FUND_LEVELS where 1=1 union all select 0,'Property' where 1=1 union all select -1,'

	where 1=1 order by 1
LevelID	SELECT illevel ValueField ,stype TextField from FUND_LEVELS where 1=1 union all select 0,'Property' where 1=1 union all select - 1,"where 1=1 order by 1
NumberFormat	1
NumberFormat	100
NumberFormat	1K
NumberFormat	1M
ReportingCurrency	Investment
ReportingCurrency	Investor
ReportStyle	Investment
ReportStyle	Investor
ReportType	Capital Roll Forward
ReportType	Transaction Detail
ShowOwnershipColumn	1
ShowOwnershipColumn	-1
ShowOwnershipColumn	True

## IM Custom Financial Data

IsConsolidate	1
IsConsolidate	False
IsConsolidate	TRUE
ReportStyle	investmentview
SuppressZero	0
SuppressZero	1
SuppressZero	TRUE

## IM Data

ColumnsToDisplay	AllColumnsFromTemplate
ColumnsToDisplay	BeginningMarketValue
ColumnsToDisplay	CapitalBalance
ColumnsToDisplay	Commitment
ColumnsToDisplay	Contribution
ColumnsToDisplay	Distribution
ColumnsToDisplay	FundedEquity
ColumnsToDisplay	InvestorFirstCommitmentDate
ColumnsToDisplay	NoOfShares
ColumnsToDisplay	OwnerShipPercent
ColumnsToDisplay	ROC
ColumnsToDisplay	UnFundedEquity
DateRange	PTD
InvestorCodes	SELECT dbo.CBREGI_InvestmentCodes(N'Joint Venture (Asset Level)', N'#ProductCode#')

InvestorCodes	SELECT Dbo.CBREGIUS_GetProductParams(N'#ProductCode#',N'entity',N'',
InvestorCodes	SELECT dbo.CPR20619_Propfilter('#InvestorID#')
InvestorCodes	SELECT stuff((SELECT '^'+rtrim(f.scode) FROM AIP_Custom_Get_Funds f WHERE f.hFund IN (#hFund#)AND hEntity IN(select item from dbo.fnSplit('#hlegal#',',')) union all Select hentity from AIP_Custom_Get_Funds where '#hlegal#'='')FOR XML path(")),1,1,")
LevelCode	Asset
LevelCode	Joint Venture (Asset Level)
LevelCode	PROPERTY
LevelCode	SELECT ilevel ValueField ,stype TextField from FUND_LEVELS where 1=1 union all select 0,'Property' where 1=1 union all select - 1,NULL where 1=1 order by 1
LevelID	0
LevelID	-1
ReportingCurrency	investmentview
ReportStyle	InvestmentView
ReportStyle	Investorview

## IM Performance Data

PerfField	A_BEGINNINGMARKETVALUE
PerfField	dCashBalance
PerfField	dForeCastGain
PerfField	dForeCastIRR
PerfField	dSharePrice
PerfTable	ASCAP_AssetInfo
ReportStyle	investmentview

## Inspection Analytics DataTable

EntityTypeFor	Property
ReportName	InspectionSummary

## Inspection Analytics SQL

DetailType	Charge
DetailType	Detail
DetailType	Footer
DetailType	Header
DetailType	InspNow
DetailType	Photo
DetailType	TenantSignature
DisplayType	InspNow
EntityTypeFor	Asset

EntityTypeFor	Building
EntityTypeFor	DueDiligenceProperty
EntityTypeFor	DueDiligenceUnit
EntityTypeFor	Room
EntityTypeFor	Unit
InspectionTypeList	SELECT " TextField,NULL ValueField union SELECT stype TextField, stype ValueField FROM insp_InspectionType WHERE 1=1 and ISNULL(bhistorical,0) = 0
PrimaryStatus	SELECT " TextField,NULL ValueField union select sText TextField, sText ValueField from insp_Lookup where 1=1 and ilistType = 0 and ISNULL(bHistorical,0) = 0
PrintPhotos	NO
PrintPhotos	YES
ReportName	InspectionDetailsReport
ReportName	InspectionDuration
ReportName	InspectionSummary
ReportName	InspectionUnassignedDirectory
ReportName	UnCompletedInspection
SecondaryStatus	SELECT " TextField,NULL ValueField union select sText TextField, sText ValueField from insp_Lookup where 1=1 and ilistType = 1 and ISNULL(bHistorical,0) = 0
ShowFailDetailsOnly	NO
ShowFailDetailsOnly	YES
ShowFailOnly	NO
ShowFailOnly	YES
ShowTenantCharge	NO
ShowTenantCharge	YES
SortOrder	SELECT TextField, TextField ValueField from (Values(" ),('Due Date'),('Inspection ID'),('Inspection Type'),('Property Code'),('Unit Code'),('Zip Code'),('Tenant Code'),('Tenant Name')) as t(TextField) where 1=1
TemplateType	SELECT " TextField,NULL ValueField union Select sName TextField,sName ValueField from insp_TemplateHeader where 1=1 and ISNULL(bHistorical,0) =0
TemplateTypeList	SELECT " TextField,NULL ValueField union Select sName TextField,sName ValueField from insp_TemplateHeader where 1=1 and ISNULL(bHistorical,0) =0
TenantStatus	SELECT Status [TextField],Status [CodeField],Status [ValueField] from tenstatus where 1=1

## International Residential Analytics

AmountPeriodType	Annual
------------------	--------

AmountPeriodType	Monthly
AmountPeriodType	Quarterly
ReportType	Lease Expiration
ReportType	Market Rent Schedule
ReportType	Potential Rent
ReportType	Unit Availability
SummaryType	Property
SummaryType	Unit
SummaryType	UnitType

## Intl Financial Analytics

FinType	Budget Comparison
FinType	TRIAL BALANCE
Grid	1
IncludeNotes	1
IsConsolidate	1
IsCosolidate	0
SuppressZero	1
TreeLevel	1
TreeLevel	2

## Intl Transaction Registers

PaymentMethod	SELECT -1 ValueField , " TextField Union Select 0 , N'Cheque' Union Select 1 , N'Cash' UnionSelect 2 , N'EFT' Union Select 3 , N'DirectDebit' Union Select 4 , N'Draft' Union Select 5 , N'ChequeSocGen' Union Select 6 , N'PaymentPlan'
SortOn	Person
Status	All
Status	Paid
Status	Unpaid
TranDetail	1
TranDetail	SELECT Case '#ReportType#' When 'Charge' Then NULL Else 1 eND
TranType	Charge
TranType	CHEQUE
TranType	Invoice
TranType	Journal
TranType	Payable
TranType	Receipt
Unposted	Posted
Unposted	Unposted

## NMAmortForCustomYSR

ApplyIntResCapIntId	1
---------------------	---

ApplyIntResCapIntId	2
ApplyIntResCapIntId	3
NMCustomReportName	BorrowerBilling
NMCustomReportName	InvestorRemittance
NMCustomReportName	LoanBalances
NMCustomReportName	TrustTransactions

## PR Preferred Returns Analytics

ActivityType	Accrual Only
ActivityType	Both
ActivityType	Distribution Only
ReportType	Deal Listing
ReportType	Deal Listing
ReportType	Equity Multiple Report
ReportType	Hypothetical Adjustment Calculation
ReportType	Hypothetical Adjustment Calculation
ReportType	Hypothetical Calculation Detail
ReportType	Hypothetical Calculation Summary
ReportType	IRR Hurdle Report
ReportType	IRR Hurdle Report
ReportType	Periodic Calculation Detail
ReportType	Periodic Calculation Detail
ReportType	Periodic Calculation Rules Setup
ReportType	Periodic Calculation Rules Setup
ReportType	Periodic Calculation Summary
ReportType	Periodic Calculation Summary
ReportType	Special Distribution
ReportType	Special Distribution
ReportType	Special Distribution Rules Setup
ReportType	Special Distribution Rules Setup

## Residential Analytics

ReportType	Resident Activity
ReportType	Traffic Detail Report
ReportType	Unit Availability
SummaryType	Agent
SummaryType	None
SummaryType	Property
SummaryType	Resident
SummaryType	Source
SummaryType	Unit
SummaryType	UnitType

## Senior Community Analytics

Groupby	<pre>SELECT 'byComm' ValueField,'Community' TextField,1 AS SortBy UNION ALL SELECT 'byClc','Care Level',2 UNION ALL Select ISNULL(sName,"") ValueField,ISNULL(sName,""),3 FROM AttributeName WHERE 1=1 AND iFileType = 3 ORDER BY 3</pre>
Groupby	<pre>SELECT 'Community' ValueField,'Community' TextField,1 AS SortBy UNION ALL SELECT 'Care Level','Care Level',2 UNION ALL Select ISNULL(sName,"") ValueField,ISNULL(sName,""),3 FROM AttributeName WHERE 1=1 AND iFileType = 3 ORDER BY 3</pre>
GroupName	<pre>SELECT 'Community' ValueField,'Community' TextField,1 AS SortBy UNION ALL SELECT 'Care Level','Care Level',2 UNION ALL Select ISNULL(sName,"") ValueField,ISNULL(sName,""),3 FROM AttributeName WHERE 1=1 AND iFileType = 3 ORDER BY 3</pre>
PeriodType	Monthly
PeriodType	Weekly
ReportType	AgingByAccounting
ReportType	ConversionReport
ReportType	LeadSource
ReportType	RentRoll
ReportType	ResidentDays
ReportType	TrendActivity
ReportType	TrendOccupancy
ReportType	UnitAvailability
ReportType	WeeklyConversionRatio
Summarizeby	<pre>SELECT 'ByCommunity' AS ValueField,'Community' AS TextField UNION SELECT 'ByResident','Resident' UNION SELECT 'ByPayer','Payer' UNION SELECT 'ByChargeCodes','Charge Code' UNION SELECT 'ByUnitType','Unit Type'</pre>
Summarizeby	<pre>SELECT 'Community' TextField, 'ByCommunity' ValueField,1 MyOrderBy UNION SELECT 'Care Level' Text, 'ByCareLevel' Value,2 MyOrderBy UNION SELECT 'Resident Status' Text, 'ByResidentStatus' Value,3 MyOrderBy UNION SELECT 'Privacy Level' Text, 'ByPrivacyLeve</pre>
Summarizeby	<pre>SELECT 'Community' TextField, 'ByCommunity' ValueField,1 MyOrderBy UNION SELECT 'Care Level' Text, 'ByCareLevel' Value,2 MyOrderBy UNION SELECT 'Unit' Text, 'ByUnit' Value,3 MyOrderBy UNION SELECT 'Unit Type' Text, 'ByUnitType' Value,4 MyOrderBy ORDER BY</pre>
Summarizeby	<pre>SELECT 'Community' TextField, 'ByCommunity' ValueField,1 MyOrderBy UNION SELECT 'Care Level' Text, 'ByCareLevel' Value,2 MyOrderBy UNION SELECT 'Unit Type' Text, 'ByUnitType' Value,3</pre>

	MyOrderBy ORDER BY MyOrderBy
Summarizeby	<pre>SELECT 'Community' TextField, 'ByCommunity' ValueField,1 MyOrderBy UNION SELECT 'Market Area' Text, 'ByMarketArea' Value,2 MyOrderBy UNION SELECT 'Care Level' Text, 'ByCareLevel' Value,3 MyOrderBy UNION SELECT 'Source' Text, 'ByLeadSource' Value,4 MyOrder</pre>
Summarizeby	<pre>SELECT 'Community' TextField, 'ByCommunity' ValueField,1 MyOrderBy UNION SELECT 'Sales Counselor' Text, 'BySalesCounselor' Value,2 MyOrderBy ORDER BY MyOrderBy</pre>
Summarizeby	<pre>SELECT 'Community' TextField, 'ByCommunity' ValueField,1 MyOrderBy UNION SELECT 'Sales Counselor' Text, 'BySalesCounselor' Value,2 MyOrderBy UNION SELECT 'Care Level' Text, 'ByCareLevel' Value,3 MyOrderBy ORDER BY MyOrderBy</pre>
Summarizeby	<pre>SELECT 'Community' TextField, 'ByCommunity' ValueField,1 MyOrderBy UNION SELECT 'Sales Counselor' Text, 'BySalesCounselor' Value,2 MyOrderBy UNION SELECT 'Care Level' Text, 'ByCareLevel' Value,3 MyOrderBy UNION SELECT 'Week' Text, 'ByWeek' Value,4 MyOrde</pre>
Summarizeby	<pre>SELECT 'Community' TextField, 'ByCommunity' ValueField,1 MyOrderBy UNION SELECT 'Source Type' Text, 'BySourceType' Value,2 MyOrderBy UNION SELECT 'Source' Text, 'ByLeadSource' Value, 3 MyOrderBy ORDER BY MyOrderBy</pre>

## SFH Analytics Data Table

DetailType	Detail
DetailType	Summary
SummaryType	Home
SummaryType	Market
SummaryType	Submarket

## Tenancy Schedule

Amendments	1
Amendments	No
Amendments	Yes
ChargeSchedule	No
ChargeSchedule	Yes
MainQuery	No
MainQuery	Yes
ProgressStatus	Active
ProgressStatus	InProgress
Proposal	No
Proposal	Yes
RecoverySchedule	No

RecoverySchedule	Yes
RentSchedule	No
RentSchedule	Yes
ReportType	TenancySchedule1
RetailSchedule	No
RetailSchedule	Yes
ShowFutureActiveLease	No
ShowFutureActiveLease	Yes
ShowPendingAmendments	No
ShowPendingAmendments	Yes
Spaces	No
Spaces	Yes
VacantUnits	No
VacantUnits	Yes
valuationtype	SELECT sCode TextField, valtype ValueField FROM COMMVALUATIONTYPE

# Index

## A

ABF Valuations, 65  
 aliases, 13, 42, 47, 109  
 analytics report, 13, 14, 41, 77, 108  
 attachments, 2, 106

## B

batch number, 65  
 budget, 65

## C

caution symbol, xi  
 class name, 148, 151  
 classic view, 61  
 closing balance, 207  
 column names, 42  
 conditional format rules, 263  
 consolidated versus unconsolidated, 30  
 consolidation, 30, 32, 33  
 contacts, 123, 124, 129  
 control number, 65  
 custom correspondence, 10  
 custom filter fields, 9, 60, 82, 83, 84, 270  
 Custom Financial Analytics, 164, 165  
 custom side menu  
     adding, 158, 160  
 custom SQL query-defined filters, 72  
 custom token, 10, 47, 70, 109  
 CustomCorrespMerging, 10, 14, 279

## D

data  
     multiple rows, 13, 104  
 data column, 2, 10, 11, 70  
 data definition language, 54  
 data manipulation language, 54  
 data sources, selecting, 251  
 design considerations, 23  
 design helper  
     downloading, 248  
     setting path, 249

details, 222

digital signatures, 141  
 documentation, filter usage, 265  
 double underline, 170, 172  
 drill-down, 28, 238

## E

email, 124, 128  
 email ID, 124  
 email template, 129  
 email type, 129  
 Excel, 202, 204, 220, 233  
     hyperlinks, adding, 238  
 Excel templates  
     conditional format rules, 263  
     error checking, 262

## F

filter condition, 10, 71, 72, 103  
 filter mapping, 10, 60  
     scripted report sections, 19, 81  
     standard analytics filters, 77  
     top-level select statement, 19, 86  
 filter usage documentation, 265  
 filtration, 19, 48, 60, 63, 78, 84, 87  
 foreign key, 10, 11  
 format rules, 263  
 freeze pane, 237, 238

## G

generating reports, 6, 61, 158  
 graphs, 1, 2, 164

## H

Help, xi  
 Help buttons, xi  
 html, 112  
 hyperlinks, 238  
     creating, 258

**I**

iFilter table, 119  
 images, 111, 202, 220, 221  
 information symbol, xi  
 Investor Portal, 2, 6  
 iTypes, 271

**K**

key column, 11, 14, 18, 25, 27, 276

**L**

list  
 dynamic, 72  
 values, 68  
 lookup list  
 custom, 68, 72  
 values, 68  
 YSI, 62, 63, 64, 67

**M**

merge field, 10, 12, 108, 110  
 merge token files  
 building, 264  
 merged reports, 8, 11

**N**

nested tables, 94  
 new view, 61  
 notes  
 caution, xi  
 information, xi  
 tip, xi

**O**

online Help, xi

**P**

p.hmy in (#property#), 14, 48, 64, 82, 83, 86, 268, 272, 278  
 page break column, 9, 11, 19  
 parent-child, 65  
 PFX files, 141  
 picture, 220, 221, 222  
 pivot table, 1, 164  
 on screen, 135

primary key, 10, 11, 276

program object iTypes, 271

**Q**

QR codes, 139

**R**

relational key, 11  
 report generation screen, 27  
 report scheduler, 151  
 ad hoc reports, 137  
 report section, 10, 11, 12, 57, 58, 71, 80, 103, 255  
 report setup, 15  
 report-generation  
 screen, 9, 11, 13, 18, 19, 24, 67, 68, 72, 129, 160, 270, 276, 277  
 Reports path, 12, 19, 41, 58, 116, 173, 178, 244, 267, 275  
 roles, 123, 124  
 running balance, 207

**S**

scroll bars, 237  
 section code, 12, 51, 255, 257  
 security certificate, 141  
 select section name, 255  
 SharePoint, 2, 6, 105, 128, 129  
 show/hide details, 225  
 smart markers, 13, 256  
 basic, 203  
 creating, 254  
 enabling, 250  
 overview, 202  
 parameters, 204, 258  
 sReturnType, 45  
 standard analytics report, 42, 80, 103, 175  
 standard report, 180

**T**

t.hmyperson in (#tenant#), 83, 84, 86  
 task runner, 2, 148, 150  
 class name, 148, 151  
 parameters, 148  
 url, 148  
 templates, 6, 13, 19, 41, 49, 58, 172, 244, 256, 259, 262, 263  
 conditional format rules, 263  
 Custom Financial Analytics, 58, 164  
 email, 129  
 error checking, 262

temporary tables, 54  
 tenant, 65, 70, 82, 83, 84, 124, 129, 176, 257, 272  
 tip symbol, xi  
 token files  
     building, 264  
 tokens, 47, 60, 81, 82, 104, 268, 269, 272, 278  
     invalid, 110  
     named, 81, 82, 84, 87, 272  
     numbered, 81, 82, 83, 84  
     session and user-related, 52  
 top-level select statement, 13, 24, 27, 51, 269, 276  
 troubleshooting, 58

**U**

unique identifier, 11, 14, 18, 19, 74, 75, 276

**V**

Val 1  
 parameter, iv, 64, 65, 81, 82, 83, 84, 85, 86, 87, 88, 272  
 Voyager  
 analytics, 1, 2, 9, 14, 16, 19, 41, 42, 48, 60, 61, 69, 77, 103, 1  
 09, 242, 251, 253  
     discovering field aliases, 109

**W**

wizard, 4  
 Word  
     merge tokens, 264  
     template setup example, 93

**Y**

Yardi Excel Add-In, 242, 243, 267, 268  
 design helper  
     downloading, 248  
     setting path, 249  
 downloading, installing, 245  
     filter section, 271  
 ysiPropertyListLookup, 62, 63, 64, 65  
 ysiPropertyLookup, 62, 63  
 ysiPropertyOrDCFPropertyLookup, 62, 63  
 ysiPropertyOrListLookup, 62, 63  
 YSR Analytic Filter Usage, 265  
 YSR design helper  
     downloading, 248  
     setting path, 249  
 YSR Hyperlink Designer  
     creating, 258

YSR report filter, 9, 10, 14, 17, 19, 60, 62, 65, 69, 85, 87  
 YSR Smart Marker Designer, 242, 250, 256  
     creating, 254  
     enabling, 250  
 YSROptions, 222  
 YSRReportHistory table, 119, 132