

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

© 2017 by Yardi Systems, Inc. All rights reserved.

This product or document is protected by copyright, trademark and other intellectual property laws. Use of the product or 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. NOTHING IN THIS DOCUMENTATION IS INTENDED TO SUPERSEDE ANY EXPRESS WRITTEN AGREEMENTS PROVIDED BY YARDI SYSTEMS FOR PORTIONS OF ANY YARDI SYSTEMS PRODUCT OR SERVICE OTHER THAN THIS DOCUMENTATION. No part of this product or document may be disclosed or reproduced in any form by any means without the prior written authorization of Yardi Systems, Inc. This product or document contains proprietary information about software processes, algorithms, and data models which is Confidential Information, and constitutes trade secrets, and which is intended for utilization solely in connection with Yardi software licensees' use of Yardi software and for no 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.

TRADEMARKS

Yardi®, Yardi Systems, Inc., the Yardi logo, Beacon, CHECKscan, COLLECTplus, CONDOCAFÉ, Conductor, ENERGYplus, Enterprise, Executive Dashboard, Genesis, Genesis², HOUSINGCAFÉ®, Insight, LOBOS®, Orion®, PAYplus, PAYscan, Point2®, Procure to Pay, PROCUREplus, PropertyShark®, rCash®, RENTCAFÉ®, RENTmaximizer, SENIORCAFÉ, Sentinel, SiteStuff®, VendorCAFÉ®, VisualHOMES®, Voyager®, Yardi Energy Solutions®, Yardi® Commercial LeasingPad, Yardi® Concierge, Yardi® Corporate LeasingPad, Yardi® CRM, Yardi® Marketplace, Yardi® Military LeasingPad, Yardi® MILITARYCAFÉ, Yardi® Portal, Yardi® Report Runner, Yardi® Service Manager, Yardi® Student LeasingPad, Yardi® STUDENTCAFÉ, Yardi® Task Runner, yCheck, and yXfer are trademarks or registered trademarks of Yardi Systems, Inc. in the United States and may be protected as trademarks in other countries.

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_q

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
January 2017	q	Core 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	Core 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	Core 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	Core Correspondence Plug-in 4 and updates to the Yardi Excel Add-In	
May 2014	g	Yardi Excel Add-In	
April 2014	f	Core Correspondence Plug-in 3.1	
March 2014	e		Added chapter on working with smart markers in Excel.

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

Contents

About Yardi Voyager Yardi Spreadsheet Reporting	ix
About the Documentation	ix
Notes.....	x
Help.....	x
Chapter 1: YSR Overview	1
Introduction to YSR	1
Quick Start to Setting up YSR Reports	3
Report Production and Delivery Options	4
Publishing to PDF, Excel, and Word.....	4
Publishing to Screen.....	5
Publishing Graphs, Charts, Tabs, and Formatted Headers to Screen.....	7
Merging Reports	10
YSR Definitions	11
YSR Deliverables.....	16
Setting up YSR Deliverables	16
Chapter 2: YSR Report Setup	18
Report Setup Overview	18
Adding a YSR Report	19
Top-Level Select Statements	23
Identifying Key Columns	25
Organizing Results on the Report Generation Screen	26
Voyager Analytics Sub-Reports	27
Adding Voyager Analytics Sub-Reports	28
Adding Extra Columns of Data to Voyager Analytics Sub-Reports	31
Custom SQL Sub-Reports.....	34
Adding Custom SQL Sub-Reports	34
YSR Scripting Conventions	37
Select Statements, Named and Unnamed.....	37
Session and User-Related Tokens	38
DDLs, DMLs, and Temporary Tables	39
Example Scripts.....	40
Email, Attachments, and Other Publishing Options	42
Emailing and Attaching Reports.....	43
Publishing to SharePoint.....	47

Publishing to Portal	47
Saving Reports to Alternate Report Output Path.....	48
Chapter 3: Filtering Data with YSR	49
Filtration Overview	49
YSR Report Filter Definition.....	51
YSI Property Lookup Lists.....	51
Defining YSR Report Filter Fields.....	54
Defining Custom Filter Fields for Use with Voyager Analytics	57
Defining Filter Fields for Data Range.....	57
Defining Dynamic Lists and Custom Lookups	60
Displaying User's Filter Criteria in Title of Report (Dynamic Titles)	63
Filter Mapping for Voyager Analytics Sub-Reports	64
Filter Mapping for Custom SQL Sub-Reports.....	69
Named Tokens Versus Numbered Tokens	70
Named Token Example	72
Numbered Token Example	73
Filtering the Top-Level Select Statement.....	74
Writing WHERE Clauses.....	75
Chapter 4: YSR Report Section and Template Setup	77
Report Section and Template Setup Overview	77
Report Section Example	78
Excel Template Example.....	79
Word Template Example	80
Working with Word 2013	82
Report Output Example.....	83
Report Section Setup	83
Setting up Report Sections	84
Adding Static Attachments to Report Sections.....	87
Report Template Setup	90
Identifying Field Names for Voyager Analytics Reports	90
Adding Images to Word Templates.....	93
Displaying Text Containing HTML Tags in Word Templates.....	94
Adding an Option for Users to Select From Multiple Templates	95
Chapter 5: Generating YSR Reports.....	98
Generating YSR Reports	98
Task Runner	100
Report Scheduler	104
Making YSR Reports Available to Report Scheduler.....	104

Custom Side Menus for YSR Reports in Voyager	106
Custom Side Menus for YSR Reports in VoyagerPlus.....	108
Chapter 6: Custom Financial Analytics Reports in YSR.....	110
CFA Report Setup Overview.....	112
Setting Up CFA Reports in YSR.....	112
Working with Consolidated Data.....	113
Formatting CFA Reports	114
CFA Account Tree Reports in YSR	115
Account Tree Report with Bold and Underline.....	116
Account Tree Report with Manipulated Data	121
CFA Property Portfolio Reports in YSR.....	126
Property Portfolio Report, Summarized by Attribute	127
Property Portfolio Report, Summarized by Property	131
CFA Comparison Reports in YSR	132
Property Comparison Reports in YSR	132
Attribute Comparison Reports in YSR	140
Chapter 7: Working With Smart Markers and Excel	147
Smart Marker Overview.....	147
Smart Marker Parameters	149
Supported Parameters.....	150
Using Parameters to Add Data in Alternating Rows	150
Formula Smart Markers	151
Dynamic Formulas	152
Repeating Dynamic Formulas.....	152
Grouping and Subtotaling.....	153
Grouping Parameters.....	153
Calculating Subtotals for a Single Data Group	154
Calculating Subtotals for Multiple Data Groups.....	155
Adding Labels to Subtotals.....	157
Image Smart Markers	161
Details (Show, Hide, and Delete with YSROptions)	163
Tips and Tricks	168
Adding Special Formatting to Expanding Row Data	168
Date and Number Formatting	169
Freeze Panes.....	171
Drill-Down Links	172

Chapter 8: Yardi Excel Add-In for YSR	174
Yardi Excel Add-In Overview	174
Building YSR Report Templates with the Add-In	176
Downloading and Installing the Yardi Excel Add-In	176
Enabling the YSR Smart Marker Designer	176
Selecting Data Sources	178
Creating Smart Markers with the YSR Smart Marker Designer	183
Checking Your Excel Template for Errors	186
Building Merge Token Files for Word Templates	187
Building YSR Reports with the Add-In (Dump SQL)	189
Preparing Script Files For Use With the Yardi Excel Add-In	190
Adding a Top-Level Select Statement	190
Building the //Filter Section	191
Filter Mapping and the Value1 Parameter	194
Val1 Parameter Conventions	194
Building Report Packages (Dump SQL)	195
Appendix A: YSR Report Schema	201
Appendix B: YSR Reports and Component Files by Vertical	202
Appendix C: YSR Reports with Voyager Analytics Data Sources	245
Appendix D: Constant Values for Analytic Data Source Filter Mapping	249
Index	263

Introduction

About Yardi Voyager Yardi Spreadsheet Reporting

With Yardi Voyager Yardi Spreadsheet Reporting, you can generate and deliver presentation-quality reports using all the formatting and graphing features of MS Excel and MS Word.

This manual includes information current through Correspondence Plug-in 7.3.

System Requirements

The minimum system requirements for Yardi Spreadsheet Reporting are as follows:

Voyager 6.0	Voyager 7.0.6	Voyager 7S (70.8)
Base Compatible Version 600822SP19	Base Compatible Version 706SP6	Base Compatible Version 708_CorePI7
Correspondence Plug-in 11	Core Correspondence Plug-in 6.3	Core Correspondence Plug-in 7
Admin Utilities Plug-in 9	Admin Utilities PI 3.3	Admin Utilities Plug-in 6
YSRCorrespondence Plug-in 2.1		Commercial Plug-in 1
		Maintenance Plug-in 1
		Residential Plug-in 1t
		General Ledger AP Plug-in 2.1

About the Documentation

This document describes how to set up and use Yardi Spreadsheet Reporting.

Your system administrator customizes Yardi Spreadsheet Reporting 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 Yardi Spreadsheet Reporting. 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 Yardi Voyager Yardi Spreadsheet Reporting:

Document	Describes
<i>Voyager Core Installation and Administration Guide</i>	Installation and administration tasks common to all Voyager systems.
<i>Voyager Core Setup Guide</i>	Setup tasks common to all Voyager systems.
<i>Voyager Core User's Guide</i>	Use of features common to all Voyager systems.
<i>YSL.NET User's Guide</i>	Installation of the Yardi Excel Add-in, which offers YSR support.
<i>Yardi SQL Scripting User's Guide</i>	How to author SQL scripts, some aspects of which are applicable to YSR reports that use custom SQL data sources.
User's guide for each Voyager system or module	Installation, setup, and use of a Voyager system or module.
Procedures guides for Voyager Residential, Voyager Commercial, and Accounting	How most users typically perform common functions. Available in Microsoft Word format for user customization.
Guides for Voyager features and utilities	How to set up and use individual features, like correspondence or G/L allocations.
Guides for utilities	How to set up and use utilities, like Conductor or Voyager Workstation Administration.

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.....	3
Report Production and Delivery Options	4
YSR Definitions	11
YSR Deliverables	16

This chapter provides a basic overview of Yardi Spreadsheet Reporting (YSR) and defines the elements that make up a YSR report.

Introduction to YSR

YSR is the most flexible and customizable reporting technology that Voyager 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. YSR reports can contain two types of sub-reports:

- Reports powered by Voyager analytics.
- Reports powered by 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

To get a quick start setting up YSR reports, view the YSR videos on Client Central.

YSR Videos

How to Build a YSR Report (Part 1 of 2)	Shows how to build an example YSR report template in Excel, for use with a custom SQL-based YSR report.
How to Build a YSR Report (Part 2 of 2)	Shows how to set up a custom script-based YSR report, including defining custom filter fields and mapping filters.
Introduction to the Top-Level Select Statement	Shows how to use a top-level select statement to organize your report results.
Merged Reports and the Page Break Column	Shows how to merge reports and manipulate labels on tabbed Excel reports.

Setup Tasks

The following table provides an overview of setup tasks for a YSR report and shows where to go in the documentation to find more information.

Identify your data source.	"Introduction to YSR" on page 1.
Add a YSR report.	"Adding a YSR Report" on page 19.
Add the YSR report filter.	"YSR Report Filter Definition" on page 51.
Map filters.	Chapter 3, "Filtering Data with YSR."
Add a top-level select statement .	"Top-Level Select Statements" on page 23 and "Filtering the Top-Level Select Statement" on page 74.
Design your report template.	Chapter 4, "YSR Report Section and Template Setup."
Set up report sections.	"Report Sections Setup Screen Reference" on page 85.

Report Production and Delivery Options

In this section:

Publishing to PDF, Excel, and Word	4
Publishing to Screen	5
Publishing Graphs, Charts, Tabs, and Formatted Headers to Screen	7
Merging Reports.....	10

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.

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



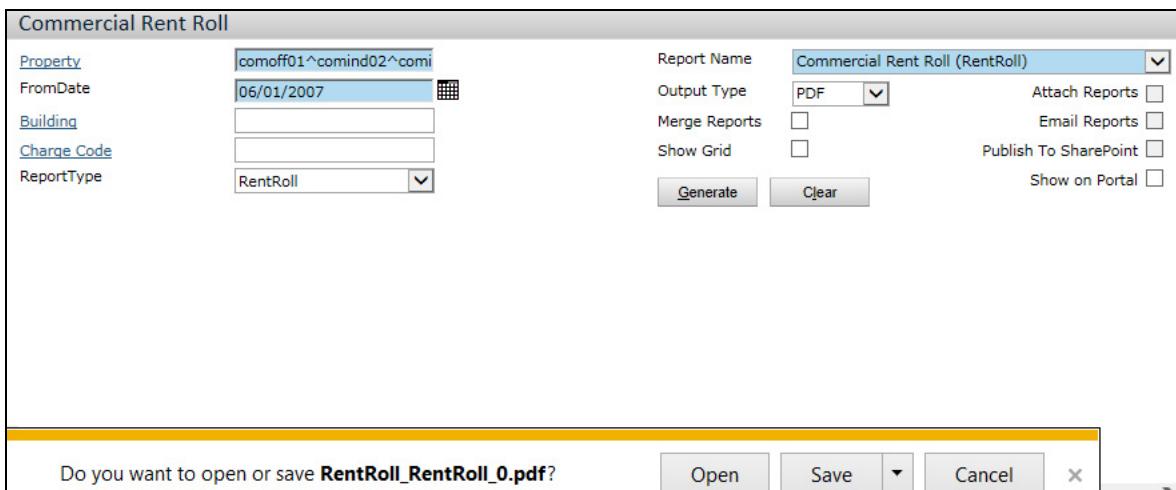
Some Excel reports with highly complex graphs or pivot tables may not have satisfactory formatting when generated on screen. In such cases, generate the report in Excel.

You can deliver all reports to contacts via email (as attachments), and you can also publish reports to SharePoint and Investor Portal. For more information about email and document management options, see “Email, Attachments, and Other Publishing Options” on page 42.

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.

investorhmy	investorcode	investorname	Report
143.00	yinv1	Investor1	View Report
144.00	yinv2	Investor2	View Report
145.00	yinv3	Investor3	View Report
146.00	yinv4	Investor4	View Report
147.00	yinv5	Investor5	View Report



For more information about organizing your report results, see “Organizing Results on the Report Generation Screen” on page 26.

Publishing to Screen

If your report contains Excel templates only, you can give users the option to publish it directly to screen. Depending on how you set up the report, users can generate a single instance of the report (for example, a rent roll for one property) or multiple instances of the same report (a rent roll for multiple properties).

- The following graphic shows a single instance of a commercial rent roll published to screen in YSR.

Commercial Rent Roll

<u>Property</u>	comoff01	<u>Report Name</u>	Commercial Rent Roll (RentRoll)
<u>FromDate</u>	06/01/2007	<u>Output Type</u>	Screen
<u>Building</u>		<u>Merge Reports</u>	<input type="checkbox"/>
<u>Charge Code</u>		<u>Show Grid</u>	<input type="checkbox"/>
<u>ReportType</u>	RentRoll	<u>Attach Reports</u>	<input type="checkbox"/>
		<u>Email Reports</u>	<input type="checkbox"/>
		<u>Publish To SharePoint</u>	<input type="checkbox"/>
		<u>Show on Portal</u>	<input type="checkbox"/>
Generate		Clear	

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

- In this example, users can generate multiple instances of the commercial rent roll by entering multiple properties in the **Property** filter field.

Commercial Rent Roll

<u>Property</u>	comoff01^comind02^comi	<u>Report Name</u>	Commercial Rent Roll (RentRoll)
<u>FromDate</u>	06/01/2007	<u>Output Type</u>	Screen
<u>Building</u>		<u>Merge Reports</u>	<input type="checkbox"/>
<u>Charge Code</u>		<u>Show Grid</u>	<input type="checkbox"/>
<u>ReportType</u>	RentRoll	<u>Attach Reports</u>	<input type="checkbox"/>
		<u>Email Reports</u>	<input type="checkbox"/>
		<u>Publish To SharePoint</u>	<input type="checkbox"/>
		<u>Show on Portal</u>	<input type="checkbox"/>
Generate		Clear	

Property	Unit(s)	Lease	Lease Type	Area	Lease From
Central Business Park,Portland	1800	VACANT	N/A		2,593,589.00
Central Business Park,Portland	1500	VACANT	N/A	88,000.00	2,593,589.00
Central Business Park,Portland	1700	VACANT	N/A	132,000.00	2,593,589.00
Central Storage,Newark	1500	VACANT	N/A	88,000.00	2,593,589.00
Central Storage,Newark	1700	VACANT	N/A	132,000.00	2,593,589.00
Central Storage,Newark	1000	General Tire Center	Industrial - Net	70,000.00	37,987.00
Central Storage,Newark	1100	Graybar	Industrial - Net	70,000.00	38,139.00
Central Storage,Newark	1200	Duncan Infrared	Industrial - Net	70,000.00	38,169.00
Central Storage,Newark	1300	Federal Express	Industrial - Net	420,000.00	38,353.00
Central Storage,Newark	1400	Aerospace Design Corporation	Industrial - Net	332,000.00	37,104.00
Sunrise Tower,Santa Monica	350A, 350B, 350C, 350D	Roxio, Inc.	Office - Net	13,380.00	37,288.00
Sunrise Tower,Santa Monica	550A, 550B	Law Offices of Wallen & Klarek	Office - Net	6,084.00	38,078.00
Sunrise Tower,Santa Monica	250A, 250B	Beeker Group Architects	Office - Net	2,484.00	38,169.00

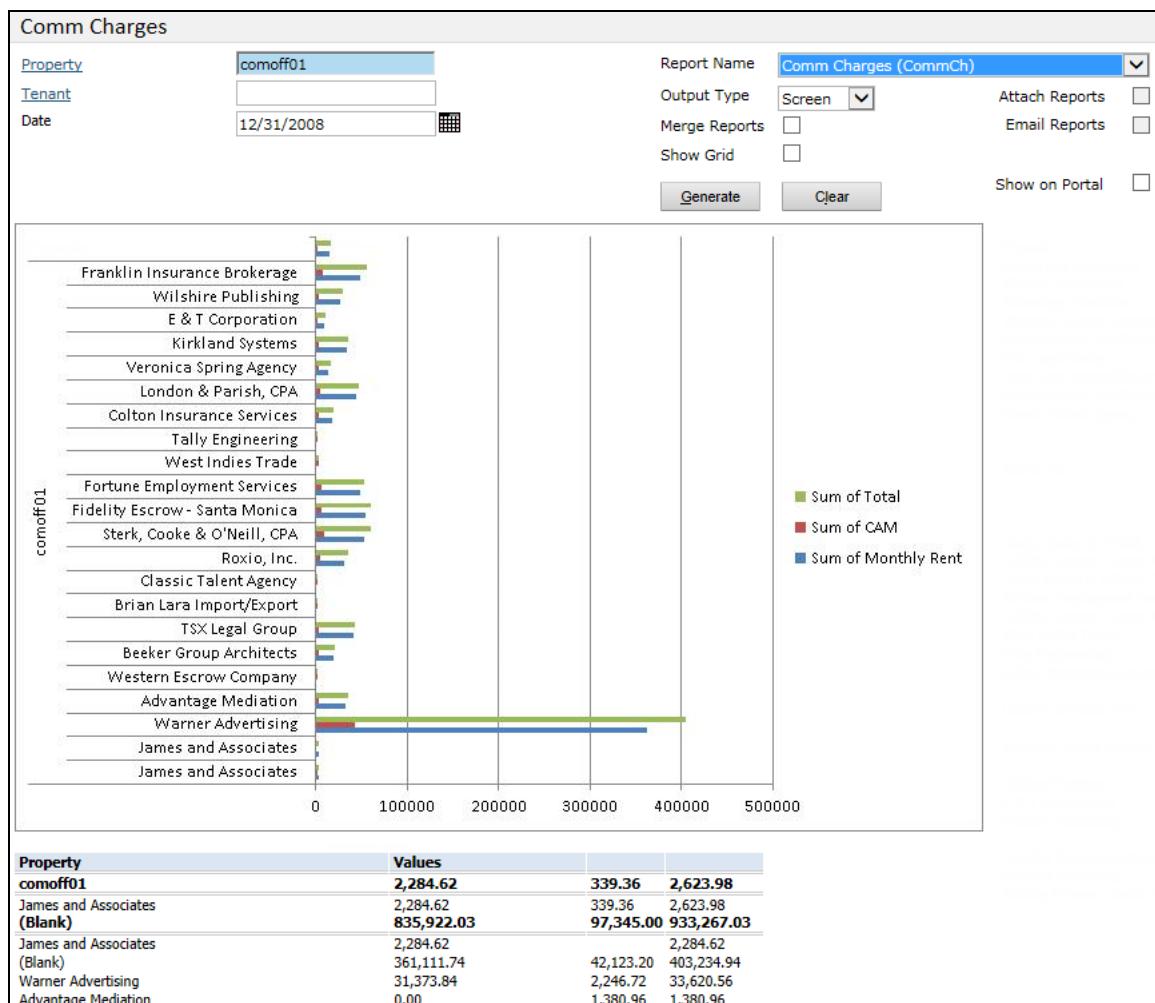
Publishing Graphs, Charts, Tabs, and Formatted Headers to Screen

With Correspondence Plug-in 5.1, you can take advantage of several features of Excel report design 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 **Merged Report Setup** screen.

The screenshot shows the 'Merged Report Setup' interface. At the top, there are fields for 'Code' (set to 'CommCh'), 'Description' ('Comm Charges'), 'Select Statement' containing SQL code, and other settings like 'Key Column' ('property') and 'Inactive'. Below these are buttons for 'Save', 'New', and various reporting options like 'Define Filters', 'Map Filters', 'Dump SQL', 'Dump Oracle', and 'Delete Setup'. A 'Notes' area is also present. At the bottom, a navigation bar includes tabs for 'Report Setup', 'Attachment & Email', 'Additional Roles', and 'Output Options'. The 'Output Options' tab is active, displaying a section titled 'Screen Output Options' with three checked checkboxes: 'Enable New Screen Output', 'Enable Tabbed Screen Output', and 'Use Freeze-Pane Setting for Report Title/Header'.

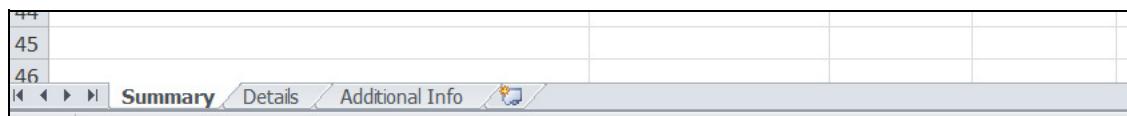
Pivot tables and graphs on screen

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:

The screenshot shows the 'Comm Charges' report configuration window. It includes fields for Property (comoff01), Tenant, and Date (12/31/2008). On the right, there are options for Report Name (Comm Charges (CommCh)), Output Type (Screen), Attach Reports, Merge Reports, Show Grid, Generate, Clear, and Show on Portal. Below the configuration is a table with tabs for Summary, Details, and Additional Info. The table displays data for various tenants with columns for Property, Tenant, Date, Monthly Rent, CAM, and Total.

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	1,380.96	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 window with a bold header row labeled 'ABC Corporation Header'. The table below has a bolded first row for the header. The columns are labeled: Property, Tenant, Date, Monthly Rent, CAM, and Total.

ABC Corporation Header					
Property	Tenant	Date	Monthly Rent	CAM	Total
comoff01	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

Merging Reports

If users are generating multiple instances of a report, they can merge them and publish them in a single Excel or PDF document by selecting the **Merge Reports** check box on the report-generation screen. For example, the following graphic shows multiple Investment Management reports merged into one Excel document. Each report appears on a separate tab.



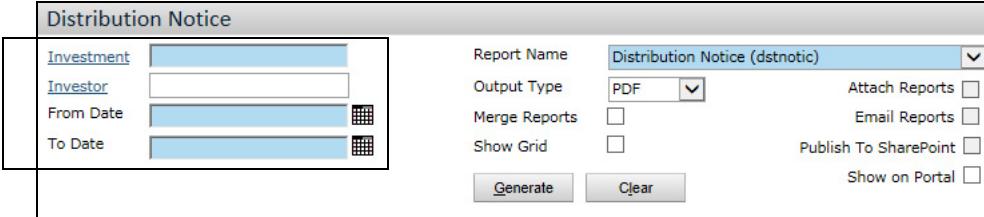
In order to display one report per tab, you must use a top-level select statement and identify the key column for every report section.



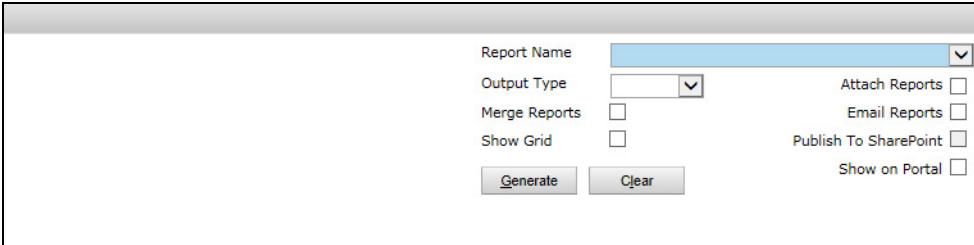
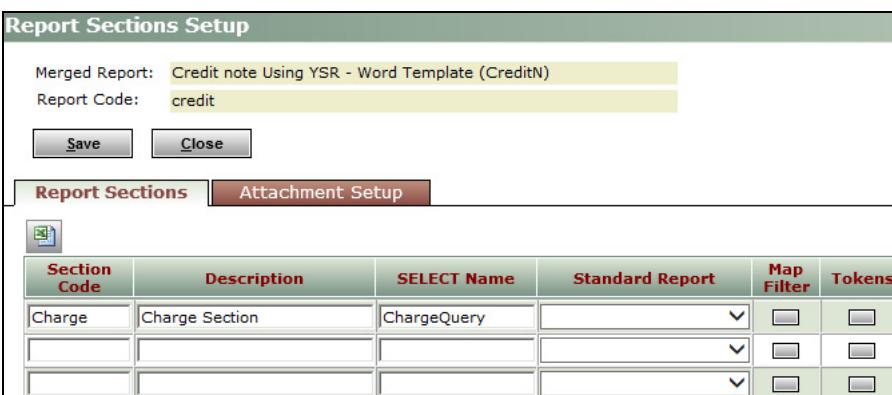
In this example, the Excel report template has unnamed worksheets. The tab names, therefore, are a combination of the report code and the page break column.

YSR Definitions

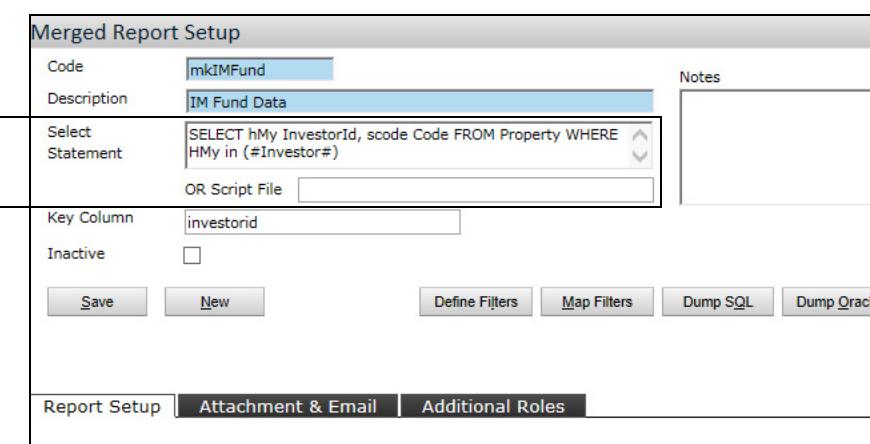
This section defines the terms used in the document to refer to various YSR elements.

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 Define Filters on the Merged Report Setup screen.</p> <p>For more information, see “Defining YSR Report Filter Fields” on page 54.</p>
Custom Correspondence	Another name for YSR. Appears on some YSR screens, such as the Custom Correspondence Setup Screen , which you can interpret as the YSR Setup Screen . A related term (CustomCorrespMerging) is the name of the root table for YSR reports in your database. Standard Voyager menu sets mix the two nomenclatures and refer to YSR Correspondence.
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 sub-report” on page 28.
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 Merged Report Setup 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>
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 (rare) or you can leave your script open and use YSR filter mapping screens to manage filter conditions. For more information about filtration, see Chapter 3, “Filtering Data with YSR.”
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 h 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>

Term	Definition
key column	See <i>primary key</i> .
list	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 60.
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 <code>ysiPropertyOrListLookup</code>, 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 60.</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 80.</p>
Merged Report	<p>Another name for YSR, which highlights the fact that a YSR report can contain many merged sub-reports. Appears on some YSR screens, such as the Merged Report Setup screen, which you can interpret as the YSR Setup Screen.</p>
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"> • If a template has named worksheets: WorksheetName-KeyColumnName • If a template has unnamed worksheets: ReportCodeValue-KeyColumnName <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>
relational key	See <i>primary key</i> .

Term	Definition																								
report code	<p>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 Code field of the Merged Report Setup 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. You can move a YSR report by clicking Dump SQL to download a PKG file.</p>																								
report-generation screen	<p>Screen where users can generate YSR reports. The screen does not have a screen name until the user selects a YSR report from the Report Name drop-down list. Then Voyager provides the name of the YSR report as the screen name.</p>																								
	<p>System Administration menu: Admin > YSR Correspondence > Generate Report</p> <p>DBO Manager screen: YSR Correspondence > Generate Report</p> 																								
report section	<p>A portion of a sub-report.</p>																								
	<p>Every sub-report in a YSR report has at least one section. You can also break a sub-report into multiple report sections, corresponding to different data queries (different SQL select statements or Voyager analytics). For example, the same sub-report might contain one section that uses Commercial Analytics and another section that uses a custom SQL select statement to retrieve lease data.</p>																								
	<p>For every report section, you must supply a section code for use with your smart markers (Excel) or merge fields (Word). If the YSR report contains more than one sub-report section, you must also define a key column to link the report section with other sections.</p>																								
	 <table border="1" data-bbox="496 1474 1388 1622"> <thead> <tr> <th data-bbox="507 1480 605 1501">Section Code</th><th data-bbox="605 1480 801 1501">Description</th><th data-bbox="801 1480 997 1501">SELECT Name</th><th data-bbox="997 1480 1192 1501">Standard Report</th><th data-bbox="1192 1480 1290 1501">Map Filter</th><th data-bbox="1290 1480 1388 1501">Tokens</th></tr> </thead> <tbody> <tr> <td data-bbox="507 1501 605 1522">Charge</td><td data-bbox="605 1501 801 1522">Charge Section</td><td data-bbox="801 1501 997 1522">ChargeQuery</td><td data-bbox="997 1501 1192 1522"><input type="button" value="▼"/></td><td data-bbox="1192 1501 1290 1522"><input type="button" value=""/></td><td data-bbox="1290 1501 1388 1522"><input type="button" value=""/></td></tr> <tr> <td data-bbox="507 1522 605 1543"></td><td data-bbox="605 1522 801 1543"></td><td data-bbox="801 1522 997 1543"></td><td data-bbox="997 1522 1192 1543"><input type="button" value="▼"/></td><td data-bbox="1192 1522 1290 1543"><input type="button" value=""/></td><td data-bbox="1290 1522 1388 1543"><input type="button" value=""/></td></tr> <tr> <td data-bbox="507 1543 605 1564"></td><td data-bbox="605 1543 801 1564"></td><td data-bbox="801 1543 997 1564"></td><td data-bbox="997 1543 1192 1564"><input type="button" value="▼"/></td><td data-bbox="1192 1543 1290 1564"><input type="button" value=""/></td><td data-bbox="1290 1543 1388 1564"><input type="button" value=""/></td></tr> </tbody> </table>	Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Charge	Charge Section	ChargeQuery	<input type="button" value="▼"/>	<input type="button" value=""/>	<input type="button" value=""/>				<input type="button" value="▼"/>	<input type="button" value=""/>	<input type="button" value=""/>				<input type="button" value="▼"/>	<input type="button" value=""/>	<input type="button" value=""/>
Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens																				
Charge	Charge Section	ChargeQuery	<input type="button" value="▼"/>	<input type="button" value=""/>	<input type="button" value=""/>																				
			<input type="button" value="▼"/>	<input type="button" value=""/>	<input type="button" value=""/>																				
			<input type="button" value="▼"/>	<input type="button" value=""/>	<input type="button" value=""/>																				
	<p>For more information, see “Report Sections Setup Screen Reference” on page 85.</p>																								
section code	<p>A short code for a select section name, used in an Excel smart marker or a Word merge field. You must map the section codes in your templates to the corresponding select section names in the data sources (the custom SQL scripts or Voyager analytics) of your report. For more information, see “Report Sections Setup Screen Reference” on page 85.</p>																								

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. Then, you must associate each named select section with a YSR report section on the Report Sections Setup 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 Merged Report Setup 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>
select statement	Portion of a SQL script that retrieves a set of records from one or more data tables.
smart marker	<p>Placeholder for Voyager data in an Excel template. Smart markers begin with &= followed by the select section code and the field alias used to retrieve the data. For example, &=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 structures. For more information, see “YSR Scripting Conventions” on page 37.</p>
standard report	<p>Appears on the Report Sections Setup screen; refers to Voyager analytics reports and a few additional data sources like IM data and Performance data.</p>
	<p>NOTE Do not confuse standard reports, as used on YSR screens, with Voyager standard reports. Voyager standard reports have limited filter options and appear in a new browser tab or window. In contrast, Voyager analytics reports have extensive report filters and display data in the same window as the report filter.</p>
sub-report	<p>Individual report contained in a YSR report. Every YSR report contains at least one sub-report, each of which references a single Excel or Word report template.</p>

Term	Definition
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 Accounts and Options screen. For more information about the Reports path, see “Web Environment” in the <i>Voyager Core Administration and Setup Guide</i>.</p>
top-level select statement	<p>Select statement that determines how results appear on the report-generation screen. If your YSR report contains more than one sub-report, the top-level select statement also contains the key column for all the scripts used in the YSR report.</p> <p>If your YSR report contains just one Excel-based sub-report, the top-level select statement is optional. (Word-based reports require a top-level select statement.)</p> <p>You can enter the top-level select statement into the Select Statement field on the Merged Report Setup screen or you can save it as a .txt file and enter the file name in the OR Script File field.</p>
	
	<p>For more information, see “Top-Level Select Statements” on page 23.</p>
Val 1 parameter	<p>Value 1 parameter. The string that follows WHERE in the WHERE clause of a SQL select statement. For example, p.hmy in (#property#) is a possible Val 1 parameter.</p> <p>For more information, see “The Value 1 Parameter” on page 69.</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 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 Report Sections Setup screen refers to Voyager analytics as Standard Reports.</p>
Yardi SQL script file	<p>A structured text file, as 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 Define Filters on the Merged Report Setup screen. For more information, see “Defining YSR Report Filter Fields” on page 54.</p>

Term	Definition
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	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.

YSR Deliverables

This section lists the deliverables that network administrators must set up for clients and describes how to load them.

Setting up 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 **Merged 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 189.

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.

- 3 Create a menu entry for the YSR report. For more information, see “Custom Side Menus for YSR Reports in Voyager” on page 106.



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.

CHAPTER 2

YSR Report Setup

In this chapter:

Report Setup Overview	18
Top-Level Select Statements	23
Voyager Analytics Sub-Reports.....	27
Custom SQL Sub-Reports	34
YSR Scripting Conventions	37
Email, Attachments, and Other Publishing Options.....	42

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 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 174.

Report Setup Overview

There are three basic elements to a YSR report:

- Report templates (Excel and Word documents).
- Data sources (SQL scripts or Voyager analytics).
- The YSR report design.

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 a YSR Report

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 specify the report templates included in the report.

After you add your YSR report, next steps include:

- Add a top-level select statement.

For information, see “Top-Level Select Statements” on page 23.

- Define the YSR report filters (custom filter fields).

For information, see “YSR Report Filter Definition” on page 51.

- Add one or more report templates.

For information, see Chapter 4, “YSR Report Section and Template Setup.”

- Map filters.

For information, see “Filter Mapping for Voyager Analytics Sub-Reports” on page 64 and “Filter Mapping for Custom SQL Sub-Reports” on page 69.

To add a YSR report

- 1 Select **Admin > YSR Correspondence > Setup Report**. The **Custom Correspondence Setup** screen appears.
- 2 Click **Submit**. The **Merged Report Setup** screen appears.

Order	Report Code	Template File	Script File	Page Break Column	Map Filter	Sections	Inactive?

3 Add a top-level select statement.



If your report contains just one Excel-based report and you want to publish it immediately to screen, this step is optional. Otherwise it is mandatory. For more information, see “Top-Level Select Statements” on page 23.

- a If you have a simple SQL query, you can enter the select statement directly into the **Select Statement** field.



Do not include //SELECT or //END SELECT. For example, you can enter a query like this:

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

- b If you have a complicated query or you want keep your query in a file rather than entering it on screen, save your query to the Reports path as a .txt file and enter the name of the file in the **Script File** field.

Leave the top-level select statement unnamed, surround it by the required //SELECT and //END SELECT strings, and make sure it is the only unnamed select in the file.

4 Complete the fields on the **Report Setup** tab. For more information, see “Merged Report Setup Screen Reference” on page 21.

5 Complete the **Output Options** tab.

Report Setup	Attachment & Email	Additional Roles	Output Options						
Screen Output Options <hr/> <table> <tr> <td>Enable New Screen Output</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Enable Tabbed Screen Output</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Use Freeze-Pane Setting for Report Title/Header</td> <td><input type="checkbox"/></td> </tr> </table>				Enable New Screen Output	<input checked="" type="checkbox"/>	Enable Tabbed Screen Output	<input type="checkbox"/>	Use Freeze-Pane Setting for Report Title/Header	<input type="checkbox"/>
Enable New Screen Output	<input checked="" type="checkbox"/>								
Enable Tabbed Screen Output	<input type="checkbox"/>								
Use Freeze-Pane Setting for Report Title/Header	<input type="checkbox"/>								

Enable New Screen Output You can generate pivot table charts and graphs on screen.

Enable Tabbed Screen Output (This check box becomes active when you select the **Enable New Screen Output** check box).

For use with tabbed Excel report templates.

Tabs appear on screen.

Use Freeze-Pane Setting for Report Title/Header (This check box becomes active when you select the **Enable New Screen Output** 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.

6 Click **Save**.

Merged Report Setup Screen Reference

Order	Report Code	Template File	Script File	Page Break Column	Map Filter	Sections	Inactive?
1	commch	CommRent.xlsx	CommRent.txt	phmy			<input type="checkbox"/>
							<input type="checkbox"/>

Select Statement	The top-level select statement. This query determines how results appear on the report-generation screen. It must contain a unique identifier of the reporting object (like a property hmy or scode), and all sub-reports must use the same key column. You must identify the unique identifier in the Key Column field. For more information, see “Top-Level Select Statements” on page 23.
OR Script File	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.
Key Column	NOTE Do not surround your statement with //SELECT or //END SELECT.
Inactive	The top-level select statement. NOTE Enter the name of the text file that contains the (unnamed) top-level select statement, or enter the whole select statement in the Select Statement field. Do not enter information to both fields.
VoyagerPlus Report	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. The top-level select statement must return this value. NOTE 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 name of the property code column (the unique identifier) here.
	Deactivates the report. Users are not able to select the report from the Report Name field on the report-generation screen, and the report setup screens are not available for editing. TIP To reactivate the report, select Admin > YSR Correspondence > Setup Report . On the Custom Correspondence Setup screen, select Yes from the Inactive? drop-down list. Locate your report, clear the Inactive check box, and save the report.

Define Filters	Opens the Report Filters Setup screen where you can set up the YSR report filter. For more information, see “Defining YSR Report Filter Fields” on page 54.
Map Filters	Opens the Filter Mapping screen where you can manage filtration for the top-level select statement. For more information, see “Filtering the Top-Level Select Statement” on page 74.
Dump SQL	Downloads a PKG file containing all the elements of the YSR report, for use in an SQL database. For more information about downloading a PKG file, see “YSR Deliverables” on page 16.
Dump Oracle	No longer supported.
Order	The merge order of the sub-reports.
Report Code	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 Report Sections Setup screen (accessible when you click the Sections button ).
Template File	The file name of the Excel or Word template Voyager uses to generate the sub-report. Make sure to save the template to Voyager’s reports path. You can include many template files in one report.
Script File	The filename of the SQL script file Voyager uses to generate this sub-report. Make sure to save the template to Voyager’s reports path. NOTE You can also use Voyager analytics, rather than a SQL script, to retrieve data and merge it with your template. If you want to use Voyager analytics only, leave this field blank.
Page Break Column	The alias of the column containing the values that Voyager uses to label merged Excel reports. If you leave this field blank, Voyager retains the worksheet name used in the Excel report template. If you supply a page break column, Voyager uses this convention: <ul style="list-style-type: none">• If a template has named worksheets: WorksheetName-KeyColumnName• If a template has unnamed worksheets: ReportCodeValue-KeyColumnName Applicable to reports that use a top-level select statement and key column only.
Map Filter 	Access point to the Filter Mapping screen corresponding to the SQL script file supporting each sub-report. Make sure that all named select statements in the script respect the filter mapping definitions established on the Filter Mapping screen, and vice versa. For more information about filter mapping, see Chapter 3, “Filtering Data with YSR.”.
Sections 	Access point to the Report Sections Setup screen. On this screen you can: <ul style="list-style-type: none">• Specify the key column for the sub-report.• Map the codes used in your Excel or Word template to the select section names used in the corresponding SQL script.• Associate report sections with Voyager analytics reports, if applicable, and complete filter mapping for those reports. For more information about using section codes in your templates, see “Report Section Setup” on page 83.
Inactive	Excludes the corresponding sub-report from the final report output.

Top-Level Select Statements

In this section:

Identifying Key Columns	25
Organizing Results on the Report Generation Screen.....	26

The top-level select statement is the select statement that appears, or can appear, in the **Select Statement** field in the top section of the **Merged Report Setup** screen. You can also save the top-level select statement in a text file in your Reports path and reference the file from the **OR Script File** field.

The screenshot shows the 'Merged Report Setup' dialog box. On the left, there is a vertical list of fields: 'Code' (set to 'CommCh'), 'Description' (set to 'Comm Charges'), 'Select Statement' (containing the SQL query 'select p.scode prop, p.saddr1 name from property p where 1=1 #condition1#'), 'OR Script File' (empty), 'Key Column' (set to 'prop'), and 'Inactive' (unchecked). On the right, there is a 'Notes' area and several buttons at the bottom: 'Save', 'New', 'Define Filters', 'Map Filters', 'Dump SQL', 'Dump Oracle', and 'Delete Setup'. A callout box labeled 'Top-level select statement' points to the 'Select Statement' field.



If you store the top-level select statement in a text file, leave it unnamed. Voyager treats the unnamed select as the top-level select. All other select statements in this text file, for potential use by your sub-reports, must have names.

Functions of the Top-Level Select

The top-level select statement performs two critical functions:

- It retrieves the primary key for the entire YSR report (often the property hmy).

The primary key is the unique identifier of a YSR report to which you must relate all sub-report data, usually by their foreign key. For example, the hproperty column is a foreign key column in the Tenant table. It contains property hmys, which makes it possible to link tenant data with property data.

You must relate the data retrieved by all sub-report select sections to the data retrieved by the top-level select by identifying the columns that contain the primary key. This means that all sub-report script files must retrieve the data column that contains the top-level primary key, and all report templates must inherit and use the same key column. 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 with a foreign key to a common parent like the property hmy. When the user generates the report, Voyager generates one report per key column value.

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.

- It determines how results appear on the report-generation screen when a user publishes results to PDF or Excel. The columns that appear on the report-generation screen are the columns retrieved by the top-level select statement.

When to Use a Top-Level Select Statement

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

- Include more than one sub-report in the report packet.
- Include attachments.
- Deliver reports via email.
- Control how results appear on the report-generation screen.
- Use one or more Word report templates.
- Generate one Excel, Word, or PDF file per key column value, rather than publishing all results immediately to screen.

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.

The top-level select statement is not supported if:

- The report draws on data sources that do not share at least one key column.

Identifying Key Columns

If your report includes more than one select statement, you must explicitly identify, for each select statement, the alias of the data column containing the primary key of report.

If you add a top-level select statement to your report, you have at least two select statements in your report (the top-level select and the select statement used in the body of the report). Therefore you must identify the aliases of their primary key.

You can identify the primary key alias of the top-level select statement by completing the **Key Column** field on the **Merged Report Setup** screen. You can identify the primary key of all sub-reports on each sub-report's **Report Sections Setup** screen, as illustrated in the following graphic. (In this example, the top-level select statement is the single unnamed select section stored in an external script file, rs_CS_OwnerListing.txt.)

Key column of the top-level SQL select statement

The screenshot displays three Yardi Voyager application windows:

- Merged Report Setup:** Shows fields for Code (qtrpkt), Description (Quarterly Report Packet), Select Statement (empty), OR Script File (rs_CS_OwnerListing.txt), and Key Column (OwnerCode). Buttons for Save, New, Define Filters, Map Filters, Dump SQL, Dump Oracle, and Delete Setup are visible. A Notes area is empty.
- Report Sections Setup:** Shows a table of report sections with columns: Order, Report Code, Template File, Script File, Page Break Column, Map Filter, and Sections. The 'Sections' column contains checkboxes. A sub-section shows Merged Report: Quarterly Report Packet (qtrpkt) and Report Code: fcs, with Save and Close buttons.
- Sub-report Sections Setup:** Shows a table of sections with columns: Section Code, Description, SELECT Name, Standard Report, Map Filter, Tokens, and Key Columns. The 'Key Columns' column contains checkboxes. A sub-section shows Section Code: budget, Description: next 12 mos budget data, SELECT Name: budget, Standard Report dropdown (selected), Map Filter checkbox, Tokens checkbox, and Key Columns (OwnerId).

An arrow points from the 'Key Column' field in the Merged Report Setup to the 'Key Columns' field in the sub-report's sections setup.

In the **Key Column(s)** fields, enter the alias used in the corresponding select section for the data column that contains the primary key of the YSR report.

Voyager generates one report per value in the **Key Column** field on the **Merged Report Setup** screen.

Organizing Results on the Report Generation Screen

In addition to establishing the primary key or relational column for all sub-reports, the top-level select statement also determines how Voyager displays results on the report-generation screen.

For example, the following graphic shows a YSR report that uses a top-level select statement to display the contact details of report recipients on the report-generation screen.

OwnerCode	Owner Name	Address	City	Phone	EmailID	OwnerId	Report
nlow0001	Abbie Greisman	Claude Debussylaan 48	Amsterdam	+31 20 301 20 22	A.Greisman@knightfrank.amsterdam.nl	2,197.00	View Report

The columns that appear above are the columns retrieved by the top-level select, and the column headers are the column aliases. Voyager adds the last column (the **Report** column) and provides a link to each report.

```
rs_CS_OwnerListing.txt - Notepad
File Edit Format View Help
//select
Select
    pr.uCode      "OwnerCode"
    ,pr.uLastName "Owner Name"
    ,pr.sAddr1    Address
    ,pr.sCity     City
    ,pr.sPhoneNum0 Phone
    ,pr.sEmail    EmailID
    ,pr.hmy      "OwnerId"
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
    ,pr.sPhoneNum0
    ,pr.sEmail
//end select
```

Voyager Analytics Sub-Reports

In this section:

Adding Voyager Analytics Sub-Reports	28
Adding Extra Columns of Data to Voyager Analytics Sub-Reports	31

Standard Voyager analytics are powerful tools for YSR reports because you can set them up without writing any custom SQL scripts. YSR setup screens refer to Voyager analytics as Standard Reports.

While you can generate analytics reports without using 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 Sub-Reports” on page 31).
- 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. (If you cannot find a template that matches one of the analytics reports, then choose another report to experiment with.) 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.

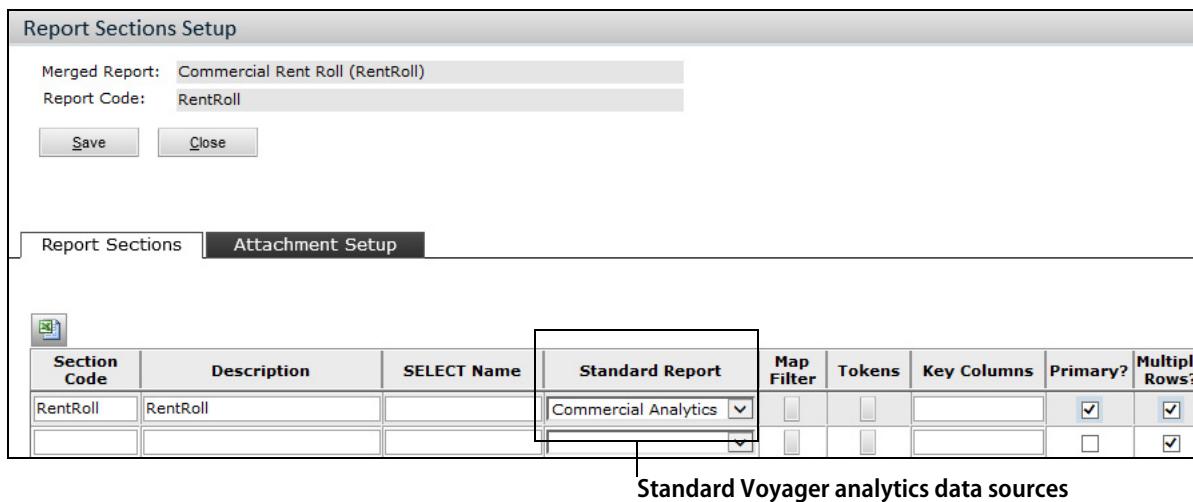
- Create a new template manually.

If you want to create your own template, read Chapter 4, “YSR Report Section and Template Setup.” This section explains how YSR maps data from specific Voyager tables and fields into report templates. See especially the section titled “Identifying Field Names for Voyager Analytics Reports” on page 90. 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 Sub-Reports

This section describes how to add a sub-report that uses Voyager analytics to retrieve data.

When you add a Voyager analytics sub-report, you select the analytics report you want to use from the **Standard Report** column on the **Report Sections Setup** screen.



The screenshot shows the 'Report Sections Setup' dialog box. At the top, there are fields for 'Merged Report' (set to 'Commercial Rent Roll (RentRoll)') and 'Report Code' (set to 'RentRoll'). Below these are 'Save' and 'Close' buttons. The main area has two tabs: 'Report Sections' (selected) and 'Attachment Setup'. Under 'Report Sections', there is a table with columns: Section Code, Description, SELECT Name, Standard Report, Map Filter, Tokens, Key Columns, Primary?, and Multiple Rows?. A row is selected in the table, and a dropdown menu is open over the 'Standard Report' cell, showing options like 'Commercial Analytics' and 'Commercial Analytics (disabled)'.

Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Key Columns	Primary?	Multiple Rows?
RentRoll	RentRoll		Commercial Analytics	<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"/>

Standard Voyager analytics data sources

If the Voyager analytics you require does not appear in the list of available analytics, 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 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 sub-report

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

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

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

- 3** Complete the **Code** and **Description** fields.

- 4** Leave all other fields on the top part of the screen blank.



The fields on the top part of the screen enable you to define a top-level SQL select statement. The top-level SQL select statement is not necessary if you are setting up a YSR report to generate just one Voyager analytics report, as in the graphic above.

If your YSR report contains multiple sub-reports, however, or if you want to control how results appear on the report-generation screen, then you must complete these fields. For more information, see “Top-Level Select Statements” on page 23.

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

Order	The display order of the reports included in the YSR report. TIP You can leave space between numbers in case you want to add reports to the report packet later.
Report Code	A code for organizing reports.
Template	The template containing the text and formatting for your report. For information about templates, see “YSR Report Section and Template Setup” on page 77.

- 6** On the **Report Setup** tab, leave the **Script File** field blank.



The **Script File** field is not necessary if you are setting up YSR to generate a report using Voyager analytics only. The **Script File** field is for use when setting up a custom YSR report that contains a section that relies on a custom SQL script.



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

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

Report Sections Setup

Merged Report: Commercial Rent Roll (RentRoll)
Report Code: RentRoll

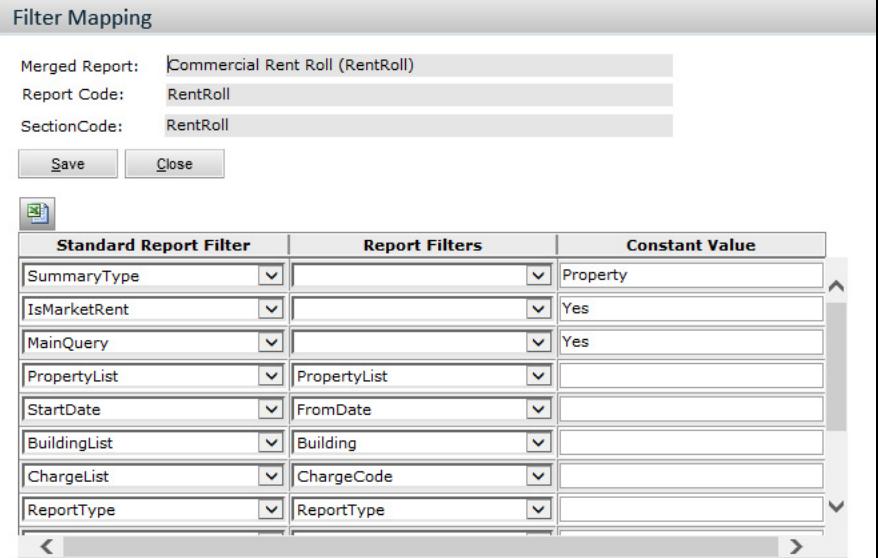
Save **Close**

Report Sections **Attachment Setup**

Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Key Columns	Primary?	Multiple Rows?
RentRoll	RentRoll		Commercial Analytics	<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"/>

- 9 Complete the fields. For more information, see “Report Sections Setup Screen Reference” on page 85.

- 10** Click the **Map Filter** button  in the row corresponding to the analytics report. The **Filter Mapping** screen appears.



Standard Report Filter	Report Filters	Constant Value
SummaryType		Property
IsMarketRent		Yes
MainQuery		Yes
PropertyList	PropertyList	
StartDate	FromDate	
BuildingList	Building	
ChargeList	ChargeCode	
ReportType	ReportType	



The custom filter fields in your YSR report filter appear in the drop-down lists in the **Report Filters** column. The report filters associated with the analytics you are using (Commercial Analytics, in this example), appear in the **Standard Report Filter** column.

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

Possible values for the **Constant Value** column appear in Appendix A, "Constant Values for Analytic Data Source Filter Mapping."

- 11** Map your custom filter fields to the standard report filters.



You can also assign a constant value to a filter by selecting an option in the **Constant Value** section.

- 12** Click **Save**.

- 13** Click **Close**.

Adding Extra Columns of Data to Voyager Analytics Sub-Reports

In some cases you may need to use YSR to add a column of data to an existing Voyager analytics report. You can add additional data columns to Voyager analytics data sources that return a string data type (as opposed to a .NET data table).

You can identify the reports that return SQL strings by reviewing 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 **Custom Correspondence Setup** screen appears.
- 2 Click **Submit**. The **Merged Report Setup** screen appears.

The screenshot shows the 'Merged Report Setup' interface. At the top, there are fields for 'Code' (RentRoll), 'Description' (Commercial Rent Roll), 'Select Statement', 'OR Script File', 'Key Column', and 'Inactive'. Below these are buttons for 'Save', 'New', and several actions: 'Define Filters', 'Map Filters', 'Dump SQL', 'Dump Oracle', and 'Delete Setup'. A 'Notes' text area is on the right. At the bottom, tabs for 'Report Setup', 'Attachment & Email', 'Additional Roles', and 'Output Options' are visible. The main area contains a table with one row:

Order	Report Code	Template File	Script File	Page Break Column	Map Filter	Sections	Inactive?
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. It displays the 'Merged Report' (Commercial Rent Roll (RentRoll)) and 'Report Code' (RentRoll). Below are 'Save' and 'Close' buttons. At the bottom, tabs for 'Report Sections' and 'Attachment Setup' are visible. The main area contains a table with one row:

Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Key Columns	Primary?	Multiple Rows?
RentRoll	RentRoll		Commercial Analytics				<input checked="" 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.

Token Name	Specify a unique name to use as the column alias.
Select Query	<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 Custom Token Setup 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 91.</p>
Inactive	Excludes the custom data from the analytics report.

- 6 Click **Save**.

- 7 Click **Close**.

Custom SQL Sub-Reports

This section describes how to add custom SQL-based sub-reports to a YSR report.

Adding Custom SQL Sub-Reports

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

You can apply filtration to every SQL select section in your script file. You impose filtration by adding either #conditionN# (where N is the sequence number of the corresponding custom filter field in the YSR report filter) to the WHERE clause, or by using (#FilterFieldName#).

If you use #conditionN#, you must also complete the **Filter Mapping** screen for the corresponding report section by adding a Val 1 parameter (the string that follows WHERE in the WHERE clause). Voyager appends (ANDs) the Val 1 parameter to your SQL script at run time.

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.

For more information and detailed examples, see “Filter Mapping for Custom SQL Sub-Reports” on page 69.

To add a custom SQL sub-report

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

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

The screenshot shows the 'Merged Report Setup' interface. At the top, there are fields for 'Code' (CommCh), 'Description' (Comm Charges), 'Select Statement' (SQL query), 'Key Column' (prop), and 'Inactive' (checkbox). Below these are buttons for 'Save', 'New', and several actions: 'Define Filters', 'Map Filters', 'Dump SQL', 'Dump Oracle', and 'Delete Setup'. A 'Notes' area is also present. At the bottom, tabs include 'Report Setup' (selected), 'Attachment & Email', 'Additional Roles', and 'Output Options'. A large table below the tabs lists report components:

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

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

Order	The display order of the reports included in the YSR report.
	TIP You can leave space between numbers in case you want to add reports to the report packet later.
Report Code	A code for organizing reports.
Template File	The template containing the text and formatting for your report. For more information about templates, see "YSR Report Setup" on page 18.
Script File	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 Filter** button corresponding to your sub-report. The **Filter Mapping** screen appears.

The screenshot shows the 'Filter Mapping' dialog box. At the top, it displays 'Merged Report: Comm Charges (CommCh)' and 'Report Code: commch'. Below this are 'Save' and 'Close' buttons. The main area is titled 'Report Filters' and contains a table with three rows of filter conditions:

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

- 6 Complete the fields as necessary. For more information, see "Filter Mapping for Custom SQL Sub-Reports" on page 69.
- 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.

The screenshot shows the 'Report Sections Setup' dialog box. At the top, it displays 'Merged Report: Comm Charges (CommCh)' and 'Report Code: commch'. Below this are 'Save' and 'Close' buttons. The main area has tabs for 'Report Sections' (which is selected) and 'Attachment Setup'. The 'Report Sections' tab displays a table with two rows of section definitions:

Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Key Columns	Primary?	Multiple Rows?
MyCharge	charges	charges		<input type="checkbox"/>	<input type="checkbox"/>	property	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MyImage	images	images		<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: &=MyCharge.ChargeDate

- 10 Complete the fields. For more information, see "Report Sections Setup Screen Reference" on page 85.

11 Click **Save**.

12 Click **Close**.

YSR Scripting Conventions

In this section:

Select Statements, Named and Unnamed.....	37
Session and User-Related Tokens	38
DDLs, DMLs, and Temporary Tables	39
Example Scripts.....	40

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.

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 and un-supported tokens.

Token	Description	Example value
#@@ATTACHREPORTS#	Returns True if the user selects the Attach Reports 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 acct, aliased AS a . TIP #@@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 acct AS b , use #@@ACCTCODE^b#.	1110-0000
#@@EMAILREPORTS#	Returns True if the user selects the Email Reports 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 Merge Reports check box on the report generation screen; False otherwise.	True
#@@OUTPUTTYPE#	Returns the user's selection from the Output Type field on the report generation screen.	EXCEL
#@@PUBLISHTOSHAREPOINT#	Returns True if the user selects the Publish to SharePoint 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 Show Grid 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.	http://yardidemo/demo708pi7

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.

```

//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 must complete both activities (creating the table and mining it) within the same named SELECT section. Create a local temporary table as a named object inside a named SELECT section, using the SQL Server convention of starting an object name with the number sign (#). The table is dropped automatically when the session goes out of scope or when your named SELECT explicitly drops it.

YSR does not support the method of creating a temporary table in the No Crystal section, mining in the Main section, and deleting it in the No Crystal After section.

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          "OwnerCode"
    ,pr.uLastName    "Owner Name"
    ,pr.hMy           OwnerID
    ,pr.sAddr1         Address
    ,pr.sCity          City
    ,pr.sPhoneNum0    Phone
    ,pr.sEmail         EmailID
    ,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
    ,pr.sPhoneNum0
    ,pr.sEmail
//end select
```

Example Script File With Two Named Select Sections (truncated)

```

//Select OpeningBalAmt
Select
    o.ucode                                         OwnerCode,
    o.ucode+'_'+p.scode+'_'+t.ssegment1          KeyValue,
    p.scode                                         PropCode,
    isnull(p.saddr1, '') + ' (' + rtrim(p.scode) + ')'  PropertyDesc,
    t.ssegment1                                     Recoverability,
    RecMast.sDesc                                    RecoveryDesc,
    Left(Convert(Varchar,Convert(DateTime,'#Date1#'),107),3)+'-'
        +Right(Convert(Varchar,Convert(DateTime,'#Date1#'),107),4)
        +' To '+Left(Convert(Varchar,Convert(DateTime,'#Date2#'),107),3)
        +'-' +Right(Convert(Varchar,Convert(DateTime,'#Date2#'),107),4)
                                                StatementPeriod,
    sum(t.sbegin)                                     OpBalAmount
From
    gltotal t
    Inner join Property p on (t.hpptyp=p.hmy)
    Inner join Owner o on (p.hlegalentity =o.hmyperson)
    Inner Join APBMasterRecoverability RecMast On RecMast.sCode=t.sSegment1
Where
    1 = 1
    and t.ibook = 0
    #Conditions#
    and t.umonth = '#Date1#'
    and t.hacct = '#AcctID#'
group by
    o.ucode,
    o.ucode+'_'+p.scode+'_'+t.ssegment1,
    p.scode,
    isnull(p.saddr1, '') + ' (' + rtrim(p.scode) + ')',
    t.ssegment1,
    RecMast.sDesc
//END select

//Select IncomeSummary
Select
    o.ucode                                         OwnerCode,
    o.ucode+'_'+p.scode+'_'+d.ssegment1          KeyValue,
    p.scode                                         PropCode,
    d.ssegment1                                     Recoverability,
    ct.sname                                         ChargeDesc,
    sum(d.samount) + sum(isnull(d.cVatAmount ,0)) GrossRcptAmt
From
    Trans tr
    Inner join detail d on (d.hInvorRec = tr.hMy)
    inner join trans trr on (trr.hmy = d.hchkorchg )
    Inner join Property p on (d.hprop=p.hmy)
    Inner join Owner o on (p.hlegalentity =o.hmyperson)
    inner join chargtyp ct on ct.hmy = trr.hretentionacct

```

Email, Attachments, and Other Publishing Options

In this section:

Emailing and Attaching Reports.....	43
Publishing to SharePoint.....	47
Publishing to Portal.....	47
Saving Reports to Alternate Report Output Path	48

This section describes the publishing options available with YSR.

Emailing and Attaching Reports

You can set up YSR to email reports and attach them to Voyager objects.

Emailing Reports



In order to email reports as attachments, you must use Voyager Workstation Administration to load the following package: ss_CustomCorrespObjTypes.pkg.



Voyager by default limits attachments to files less than 5MB in size. System administrators can override this restriction by using Voyager Workstation Administration to load the following package: optAttachmentNoFileSizeLimit.pkg.

There are three ways to tell Voyager where to email reports.

- You can enter an email address manually when setting up the report.
- You can send the report to the email ID associated directly with a person record (a tenant, employee, owner, vendor, or other person type record).
- You can send the report to any contact associated with a person record. (All contacts have roles; send the email to the contact with the appropriate role.)

For example, the following graphic shows one type of person record, an owner record. This owner has two contacts. You can choose whether to send the report to the owner's email address or to one of the owner's contacts. Contact records are organized by roles.

Legal Entity		Data	
Code	crownon	Email Address	ontcorp@ontcorp.ca
Name	Ontario Corporation Ltd.	Alternate Email	
Address	9899 Airport Road	Office	
City	Mississauga	Home	
Prov-Post Code	ON L6T 2H7		
Country	cn		
Notes	Ontario Corporation owns real estate in Ontario and manages these portfolios		
<input type="button" value="Edit"/> <input type="button" value="New"/> <input type="button" value="Close"/> <input type="button" value="Help"/>			
<input type="button" value="Properties"/> <input type="button" value="Tax Info"/> <input type="button" value="Other Info"/> <input type="button" value="Payment Info"/> <input type="button" value="Contacts"/>			
<input type="button" value="Edit"/> 			
Role	Primary	Company	Name
Executive	Y	Ontario Corp. Ltd.	Demetrius Costandinos
Billing Contact			(905) 231-3133
			Demetrius@OntarioCorp.Ltd.com
			ontcorp@ontcorp.ca

→ Email ID of owner

→ Emails of owner's contacts



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.

For more information about contacts, roles, and email templates, see the *Voyager Workflows and Notifications User's Guide*.

Attaching Reports to Voyager Objects

You can set up YSR reports so that users have the option of attaching them to the related Voyager object. For example, the following graphic shows a YSR report that is set up to attach reports to Fund objects (investment entities).

The screenshot shows the 'Merged Report Setup' interface. The 'Code' field is set to 'scapcall'. The 'Description' field contains 'Standard Capital Call Notice'. The 'Select Statement' field displays a SQL query: 'SELECT p.hmy investorid, p.scode [Investor Code], ...'. The 'Key Column' is 'investorid'. The 'Inactive' checkbox is unchecked. A note on the right side states: 'Contact information on report output is based on contact roles setup on this YSR report. Investment contact info is hard coded to that of the role "Fund Manager". Funded equity from IM template selected in filter determines previously contributed capital amount. Report only generates for active commitments, not necessarily funded.' Below the main setup are buttons for 'Save', 'New', 'Define Filters', 'Map Filters', 'Dump SQL', 'Dump Oracle', and 'Delete Setup'.

Below this is a tabbed section with 'Report Setup' selected, followed by 'Attachment & Email', 'Additional Roles', and 'Output Options'.

The 'Object Type Setup' tab shows 'Object Type' set to 'Fund' and 'ID Column' set to 'investorid'. It also includes a note: '(Set hMy column for Object : Fund)'.

The 'Attachment Setup' tab shows 'Attachment Type' as a dropdown menu. The 'Email Setup' tab includes fields for 'Email Template', 'Email Type', 'Email To (Role)', and 'Email Output To (Specify Email-ID)'.

The 'Report Output Path' tab shows 'Output Path' as a dropdown menu.

To set up your YSR report for attachment to Voyager objects, you must identify the object type and the alias of the object's key column (**ID Column**).



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.

Once you have completed the necessary setup steps, users can select the **Attach Reports** check box when generating the report:

The screenshot shows the 'Standard Capital Call Notice' report generation screen. It includes fields for 'Investor' (with value '13'), 'IM Template' (set to 'MnthStmt'), 'Report Name' (set to 'Standard Capital Call Notice (scapcall)'), 'Output Type' (set to 'PDF'), 'Attach Reports' (checkbox checked), 'Merge Reports' (checkbox unchecked), 'Show Grid' (checkbox unchecked), 'Generate' button, 'Clear' button, and 'Show on Portal' checkbox.

Voyager generates the report (or reports, if generated for multiple entities) and attaches the report to each object.

Attachments							
Available Attachments for Fund: Canada Investor (cdinv)							
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"/>	

Email and Attachment Permissions

You can set user permissions for attaching and emailing reports.

The relevant tokens are:

- Correspondence:YSR - Attach Report
- Correspondence:YSR - Email report

For more information about setting permissions, see “Menu and Data-Access Permissions” in the *Voyager Core Administration and Setup Guide*.

To set up attachments and emails

- 1 Select **Admin > YSR Correspondence > Setup Report**
- 2 Click **Submit**. The **Merged Report Setup** screen appears.
- 3 Click the **Attachments & Email** tab.
- 4 Select a data source in the **Object Type** field. The **ID Column** field becomes mandatory.
- 5 Complete the fields. For more information, see “Attachment & Email Tab Screen Reference” on page 46.



For more information about setting up email templates and attachment types, see the *Voyager Workflows and Notifications User’s Guide*.

- 6 Click **Save**.

Attachment & Email Tab Screen Reference

The screenshot displays the 'Attachment & Email' tab of a report setup interface. The tabs at the top are 'Report Setup', 'Attachment & Email' (which is selected), and 'Additional Roles'. The main area is divided into several sections:

- Object Type Setup:** Shows 'Object Type' set to 'Tenant', 'ID Column' set to 'hMyPerson' (with a note '(Set hMy column for Object : Tenant)'), and a note '(Set hMy column for Object : Tenant)'.
- Attachment Setup:** Shows 'Attachment Type' (dropdown menu), 'Email Template' set to 'Residential - Outstanding A', 'Email Type' set to 'Tenant', and 'Email To (Role)' (dropdown menu).
- Email Setup:** Shows 'Email Output To (Specify Email-ID)' (text input field).
- Report Output Path:** Shows 'Output Path' (text input field).
- Publish To SharePoint (Publish Source):** Shows 'SPPUBLISHING_AginDet_SOURCE =' (text input field).

Object Type	The data table from which the email template accesses information.
ID Column	The alias of the column that holds the unique identifier (the handle) of the data table specified in the Object Type field. TIP The value in this field should match the key column specified on the YSR report setup screens.
Attachment Type	The name Voyager gives to the file when attaching it to email.
Email Template	The email template Voyager uses. For more information about setting up email, see "E-Mail Templates" in the <i>Voyager Workflows and Notifications User's Guide</i> .
Email Type	Determines whether Voyager sends email directly to the email ID associated with a person record, or to the contacts associated with the person record. TIP The options available in this field depend on your selection in the Object Type field. All object types have contacts; some object types also have person records. If you select Contact , the Email To (Role) drop-down list becomes active. TIP To send an email to a tenant, select Tenant in the Object Type field and the Email Type field.
Email To (Role)	This field becomes active if you select Contact from the Email Type drop-down list. The role of the contact to whom Voyager sends the report. For more information about contacts, see the <i>Voyager Workflows and Notifications User's Guide</i> .
Email Output To (Specify Email-ID)	(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.
Publish To SharePoint (Publish Source)	(This field appears if you load the optional package to publish to SharePoint) 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.

Publishing to SharePoint

You can also publish YSR reports to SharePoint.



To enable publishing to SharePoint, load the following package:
Opt_ss_Intl_CorrespondenceEnablePublishToSharepoint.pkg

You must specify the SharePoint source on the Attachment & Email Tab.

If you enable the option to publish reports to SharePoint, a new check box appears on the report-generation screen.

The screenshot shows a software interface for generating reports. At the top right, there are several checkboxes: 'Attach Reports', 'Email Reports', 'Publish To SharePoint' (which is checked), and 'Show on Portal'. Below these are two buttons: 'Generate' and 'Clear'.

When you select this check box and generate the report, Voyager displays a field indicating whether the report published to SharePoint successfully.

Publishing to Portal

Portal users can publish documents to Portal directly from the YSR report generation screen.

The screenshot shows the 'Standard Capital Call Notice' report generation screen. It includes fields for 'Investor' (set to '13'), 'IM Template' ('MnthStmt'), 'Report Name' ('Standard Capital Call Notice (scapcall)'), 'Output Type' ('PDF'), 'Merge Reports', 'Show Grid', 'Generate' button, 'Clear' button, and 'Show on Portal' checkbox (which is checked). Below the report name, there is a table:

investorid	Investor Code	Investor Name	Investment Code	Investment Code1	Report
136.00	cdinv	Canada Investor	nlfund	Netherlands Fund	View Report
137.00	nlinv	Netherlands Investor	nlfund	Netherlands Fund	View Report



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 "Attaching Reports to Voyager Objects" on page 44.

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.

Attachments							
Available Attachments for Fund: Canada Investor (cdinv)							
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"/>	<input type="checkbox"/>
	scapcall_136_0	06/09/2016	scapcall_136_0_1.pdf	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Saving Reports to Alternate Report Output Path

By default Voyager writes YSR reports to the Temp folder in the Reports path. If you want to output files to a different location, you can specify the alternate file path on the **Attachment & Email** tab, in the **Output Path** field.

Report Setup	Attachment & Email	Additional Roles
Object Type Setup Object Type: Tenant ID Column: hMyPerson (Set hMy column for Object : Tenant)		
Attachment Setup Attachment Type: Email Template: Residential - Outstanding / Email Type: Tenant Email To (Role):		
Email Setup Email Output To (Specify Email-ID):		
Report Output Path Output Path: \\SharedLocation\Reports\YSRExports		



Changing the report output path does not change the attachment path location.

CHAPTER 3

Filtering Data with YSR

In this chapter:

Filtration Overview	49
YSR Report Filter Definition	51
Filter Mapping for Voyager Analytics Sub-Reports.....	64
Filter Mapping for Custom SQL Sub-Reports	69

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:

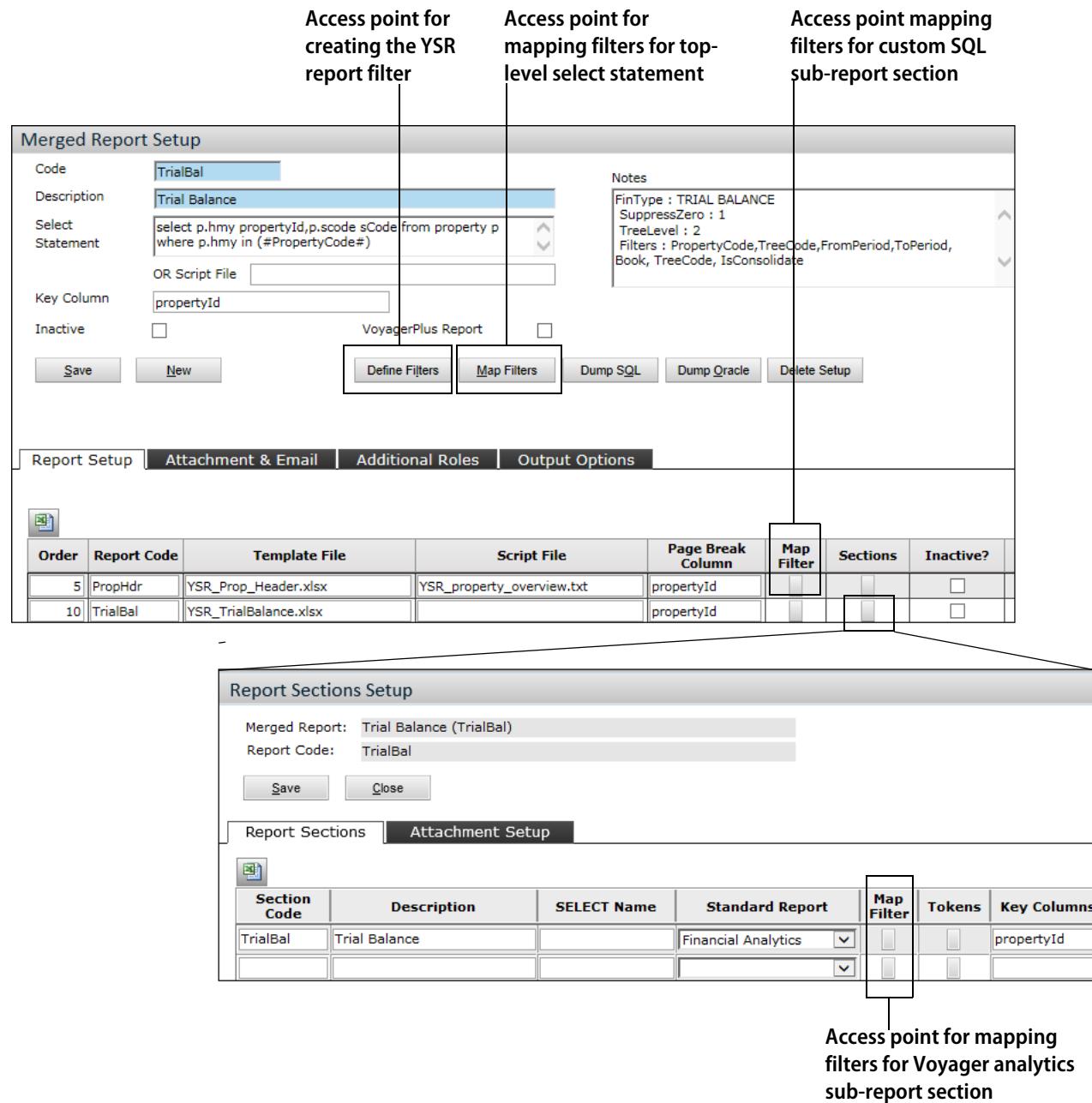
- Several types of filter technologies (YSI lookup lists, custom lookup lists, and more).
- Several types of token substitution (named tokens, numbered tokens, and session-related tokens).
- Both custom SQL scripts and Voyager analytics.
- Both hard-coded filters and filters that accept the user's input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

The following graphic provides an overview of all the access points for creating custom filter fields for your users and mapping them to the various data retrieval components of a report.



YSR Report Filter Definition

In this section:

YSI Property Lookup Lists	51
Defining YSR Report Filter Fields.....	54
Defining Filter Fields for Data Range.....	57
Defining Dynamic Lists and Custom Lookups	60
Displaying User's Filter Criteria in Title of Report (Dynamic Titles)	63

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.

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 51.

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:

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

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 Custom SQL Sub-Reports" on page 69.

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#))`

Filter Mapping

Merged Report: Comm Charges (CommCh)	Report Code: commch								
Save	Close								
<table border="1"> <thead> <tr> <th>Report Filters</th> <th>Report Filter Condition</th> </tr> </thead> <tbody> <tr> <td>property (#Condition1#)</td> <td>p.hmy in (select hproperty from listprop2 where hPropList in (#Property#))</td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table>		Report Filters	Report Filter Condition	property (#Condition1#)	p.hmy in (select hproperty from listprop2 where hPropList in (#Property#))				
Report Filters	Report Filter Condition								
property (#Condition1#)	p.hmy in (select hproperty from listprop2 where hPropList in (#Property#))								

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.

Defining YSR Report Filter Fields

Custom filter fields are the fields that appear to users at run-time. They make up the YSR report filter. When you add new custom filter fields, you can take advantage of standard Yardi filtering mechanisms like YSI lookup lists and date filters, and you can also write your own custom lookup lists.

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 add custom filter fields

- 1 Select **Admin > YSR Correspondence > Setup Report**. The **Custom Correspondence Setup** screen appears.
- 2 Complete the filter and click **Submit**. The **Merged Report Setup** screen appears.

The screenshot shows the 'Merged Report Setup' dialog box. The top section contains fields for 'Code' (CommCh), 'Description' (Comm Charges), 'Select Statement' (a SQL query), 'Key Column' (prop), and 'Inactive' (unchecked). To the right is a 'Notes' text area. Below these are buttons for 'Save', 'New', and several report-related functions: 'Define Filters', 'Map Filters', 'Dump SQL', 'Dump Oracle', and 'Delete Setup'. At the bottom, tabs for 'Report Setup', 'Attachment & Email', 'Additional Roles', and 'Output Options' are visible. A large table below the tabs shows report details with columns for Order, Report Code, Template File, Script File, Page Break Column, Map Filter, Sections, and Inactive?.

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

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

Report Filters Setup

Merged Report: Quarterly Report Packet (qtrpkt)

Save Close

Field Name	Label	Seq	Type	Lookup Name	Multi Select?	Parent	Mandatory?	List Values	Code to ID	Query?	Column View	Inactive?
Property	Property	1	Lookup List	ysiPropertyOrList1	<input type="checkbox"/>		<input checked="" type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>
Tenant	Tenant	2	Lookup List	ysiTenantLookup	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>
Date From	Date from	3	Date		<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>
Date To	Date To	4	Date		<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>
					<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>

- 4 Complete the fields.** For more information, see “Report Filters Setup Screen Reference” on page 55.
- 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

Report Filters Setup

Merged Report: Quarterly Report Packet (qtrpkt)

Save Close

Field Name	Label	Seq	Type	Lookup Name	Multi Select?	Parent	Mandatory?	List Values	Code to ID	Query?	Column View	Inactive?
Property	Property	1	Lookup List	ysiPropertyOrList1	<input type="checkbox"/>		<input checked="" type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>
Tenant	Tenant	2	Lookup List	ysiTenantLookup	<input type="checkbox"/>	1	<input checked="" type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>
Date From	Date from	3	Date		<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>
Date To	Date To	4	Date		<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>
					<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>	1	<input type="checkbox"/>

Field Name	The name of the custom filter field.
	The value that you enter here appears when you perform filter mapping.
Label	The label of the custom filter field, visible when generating a report.

Sequence	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 70 and “Numbered Token Example” on page 73. NOTE If you implement a parent-child relationship between filters, number the parent filter with a lower number than the child.
Type	The input type of the custom filter field that appears on the report-generation screen. Text Voyager accepts text values in the custom filter field. List Voyager displays a static or a dynamic list using the values you enter in the List Values field to the report. Lookup List Voyager displays a lookup list. You can select a standard YSI lookup list from the Lookup Name drop-down list, or you can create a custom lookup list using the List Values field. TIP 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. Date Voyager displays a calendar and accepts date values only. Post Month Voyager displays a calendar and accepts date values only. Check Box Voyager displays a check box.
Lookup Name	(You can use this field if you select Lookup List in the Type field.) The pre-defined lookup list that Voyager displays on the report-generation screen. NOTE If you prefer, you can enter a select query in the List Values field instead.
Multi-Select	Enables the user to select multiple values when completing the custom filter fields on the report-generation screen.
Parent	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 Tenant field (the child) depends on the Property field. TIP 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.
Mandatory	Makes the filter a required field on the report-generation screen.
List Values	Use this field to enter the values for filter fields of the List or Lookup List type. If you select List in the Type field, specify the list values separated by a caret sign (^). For example, enter residential^commercial^industrial. If you select Lookup List in the Type field, you can use this field to enter a custom select statement in lieu of selecting a pre-defined lookup list from the Lookup Name field. NOTE For more information about how to use custom SQL in this field, see “Defining Dynamic Lists and Custom Lookups” on page 60.

Code to ID	(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.
	NOTE For more information about how to use this field with lookup lists, see “Defining Dynamic Lists and Custom Lookups” on page 60.
Query	Select this check box if you are using a dynamic list or a custom lookup list.
Column View	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.
Inactive	Prevents the custom filter from appearing on the report-generation screen.

Defining Custom 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 sub-report” on page 28.

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 sub-report” on page 28. See also “Report Sections Setup Screen Reference” on page 85.

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	
<u>Property</u>	<input type="text"/>
<u>Tenant</u>	<input type="text"/>
Date From	<input type="text" value="01/01/2014"/> <input type="button" value="Calendar"/>
To	<input type="text" value="02/28/2014"/> <input type="button" value="Calendar"/>
Report Name: Commercial Charges (CommChg) <input type="button" value="▼"/> Output Type: Screen <input type="button" value="▼"/> <input type="checkbox"/> 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="button" value="Generate"/> <input type="button" value="Clear"/>	

- Linked filter range, mapped to custom SQL script

Commercial Charges	
<u>Property</u>	<input type="text"/>
<u>Tenant</u>	<input type="text"/>
Date From	<input type="text" value="01/01/2014"/> <input type="button" value="Calendar"/>
To	<input type="text" value="02/28/2014"/> <input type="button" value="Calendar"/>
Report Name: Commercial Charges (CommChg) <input type="button" value="▼"/> Output Type: Screen <input type="button" value="▼"/> <input type="checkbox"/> 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="button" value="Generate"/> <input type="button" value="Clear"/>	



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.

Report Filters Setup																																																
Merged Report: Commercial Charges (CommChg)																																																
<input type="button" value="Save"/> <input type="button" value="Close"/>																																																
<table border="1"> <thead> <tr> <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> </tr> </thead> <tbody> <tr> <td>Property</td> <td>Property</td> <td>1</td> <td>Lookup List</td> <td>ysiPropertyListLookup</td> <td><input checked="" type="checkbox"/></td> <td></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Tenant</td> <td>Tenant</td> <td>2</td> <td>Lookup List</td> <td>ysiTenantLookup</td> <td><input type="checkbox"/></td> <td>1</td> <td><input type="checkbox"/></td> </tr> <tr> <td>DateFrom</td> <td>Date From</td> <td>3</td> <td>Date</td> <td></td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td>DateTo</td> <td>Date To</td> <td>4</td> <td>Date</td> <td></td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> </tr> </tbody> </table>									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"/>
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"/>																																									

Filter-mapping On the **Filter Mapping** screen, use the logical operators \geq and \leq to complete the **Report Filter Condition** field.

The screenshot shows the 'Filter Mapping' dialog box. At the top, 'Merged Report:' is set to 'Commercial Charges (CommChg)' and 'Report Code:' is set to 'CommChg'. Below these are 'Save' and 'Close' buttons. The main area contains a table titled 'Report Filters' and 'Report Filter Condition'. The table has four rows corresponding to filter fields: 'Property (#Condition1#)', 'Tenant (#Condition2#)', 'DateFrom (#Condition3#)', and 'DateTo (#Condition4#)'. Each row shows a dropdown menu for the filter field and a text input field for the condition. The conditions are:

Report Filter	Report Filter Condition
Property (#Condition1#)	p.hmy in (#Property#)
Tenant (#Condition2#)	t.hmyperson in (#Tenant#)
DateFrom (#Condition3#)	ckd.dtdate \geq ('#DateFrom#')
DateTo (#Condition4#)	ckd.dtdate \leq ('#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, 'Merged Report:' is set to 'Commercial Charges (CommChg)'. Below these are 'Save' and 'Close' buttons. The main area contains a table titled 'Report Filters Setup'. The table has three rows with columns for 'Field Name', 'Label', 'Sequence', 'Type', 'Lookup Name', 'Multi Select?', 'Parent', and 'Mandatory?'. The first two rows have single-field names ('Property' and 'Tenant'). The third row has a linked range name ('DateFrom:DateTo') and a label ('Date From:To'). The 'Type' column for the linked range is set to 'Date'.

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:DateTo	Date From:To	3	Date		<input type="checkbox"/>		<input type="checkbox"/>

Filter-mapping On the **Filter Mapping** screen, use the logical operator *between* to complete the **Report Filter Condition** field.

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

The screenshot shows the 'Filter Mapping' dialog box. At the top, it displays 'Merged Report: Commercial Charges (CommChg)' and 'Report Code: CommChg'. Below this are 'Save' and 'Close' buttons. A 'Report Filters' section contains three entries: 'Property (#Condition1#)' with condition 'p.hmy in (#Property#)', 'Tenant (#Condition2#)' with condition 't.hmyperson in (#Tenant#)', and 'DateFrom:DateTo (#Condition3#)' with condition 'ckd.ddate between '#DateFrom#' and '#DateTo#'. There is also a 'Report Filter Condition' section below the filters.



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.

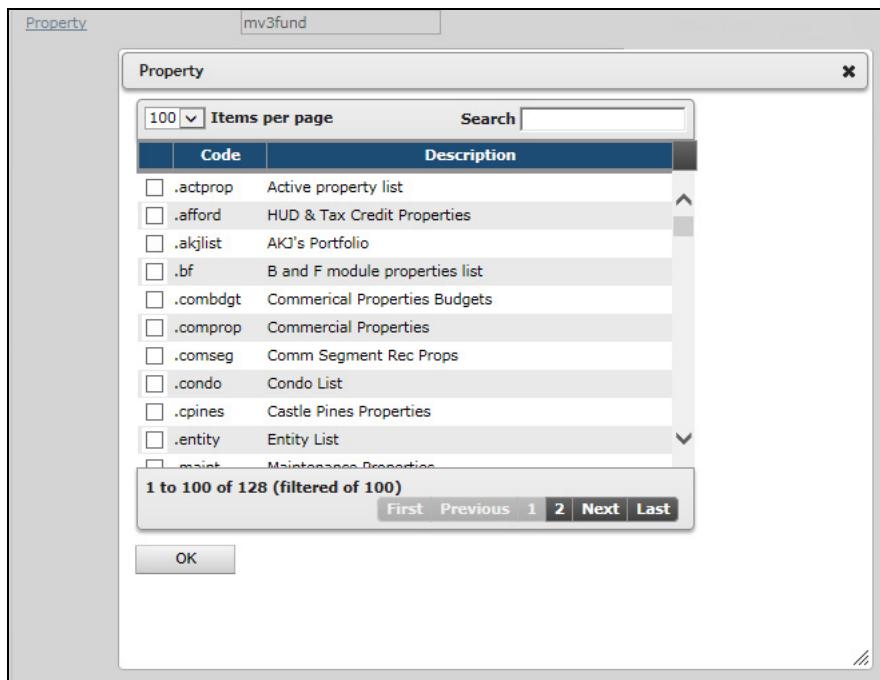
Defining Dynamic Lists and Custom Lookups

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.

The screenshot shows a dropdown menu with the label 'Currency' on the left. The list contains the following items: aud,cdn,eur,hkd,mxp,nzd,rmb,usd. The item 'aud' is highlighted with a blue selection bar.

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	61
To define a custom lookup list	62

To define a dynamic drop-down list

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

- 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 IDField 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:

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

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

Comm Charges		Report Name	Comm Charges (CommCh)	<input checked="" type="checkbox"/>
Property	comoff01	Output Type	Screen	<input type="checkbox"/> Attach Reports
Tenant	t0000016	Merge Reports	<input type="checkbox"/>	<input type="checkbox"/> Email Reports
Date	12/31/2008	Show Grid	<input type="checkbox"/>	<input type="checkbox"/> Show on Portal
		<input type="button" value="Generate"/>	<input type="button" value="Clear"/>	
Report title	Monthly Activity Report for	property: comoff01 tenant: t0000016		
	Property	Tenant	Date	Monthly Rent
	comoff01	Fidelity Escrow - Santa Monica	12/31/2008	15958.8
		Fidelity Escrow - Santa Monica	12/31/2008	15958.8
				CAM
				2291.52

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:

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

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.

Field Name	Multi Select?	Parent	Mandatory?	List Values	Code to ID	Query?	Column View	Show In Title	Inactive?
property	Pros	<input checked="" type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
tenant	Ten	<input checked="" type="checkbox"/>	1			<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
date	Dat	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>

3 Click **Save**.

4 Edit your report template:

- a If you are using Excel, add the following smart marker: &=YSRFilterValues.ReportTitleValues

A	B	C
1		
2	Monthly Activity Report for	&=YSRFilterValues.ReportTitleValues
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 Sub-Reports

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 (a standard 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.

The screenshot shows the 'Financial Reports' dialog box. It includes sections for 'Property' (with dropdowns for 'Book' and 'Account Tree'), 'Report Type' (set to 'Balance Sheet'), 'Period' (with 'From' and 'To' fields), 'Report Columns' (set to 'Actual'), and a large panel of filter fields. These filter fields include 'Asset Manager', 'Assistant Manager', 'Client', 'Country', 'Denominator', 'Department', 'Source', and 'Immediate Source'. There are also several checkboxes and dropdowns for grid options like 'Grid', 'Freeze Pane', 'Decimals', 'Show Property Name', 'Suppress Zero', 'Graph', 'Summary', 'Tree Level', 'Show Acct. Code', and 'Source'.

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:

The screenshot shows the 'Trial Balance' report configuration dialog box. It has sections for 'Property' (set to 'comoff01^comoff02'), 'Book' (dropdown), 'Account Tree' (set to 'ysi_is'), 'From Period' (set to '01/2007'), 'To Period' (set to '12/2007'), and 'Consolidate' (checkbox). On the right, there are report settings: 'Report Name' (set to 'Trial Balance (TrialBal)'), 'Output Type' (set to 'Excel'), 'Merge Reports' (checkbox), 'Show Grid' (checkbox), 'Generate' (button), 'Clear' (button), 'Attach Reports' (checkbox), 'Email Reports' (checkbox), and 'Show on Portal' (checkbox).

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.

The screenshot shows the 'Report Filters Setup' dialog box. It displays a table of filter fields for the 'Trial Balance (TrialBal)' report. The table columns are: Field Name, Label, Sequence, Type, Lookup Name, Multi Select?, Parent, Mandatory?, and List Value. The rows represent the following fields:

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	List Value
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 **Filter Mapping** screen for the YSR Trial Balance report.

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

- The Financial Analytics is hard-coded to generate the Trial Balance report type only (row 1).
- The **Property**, **Book**, **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 A, "Constant Values for Analytic 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

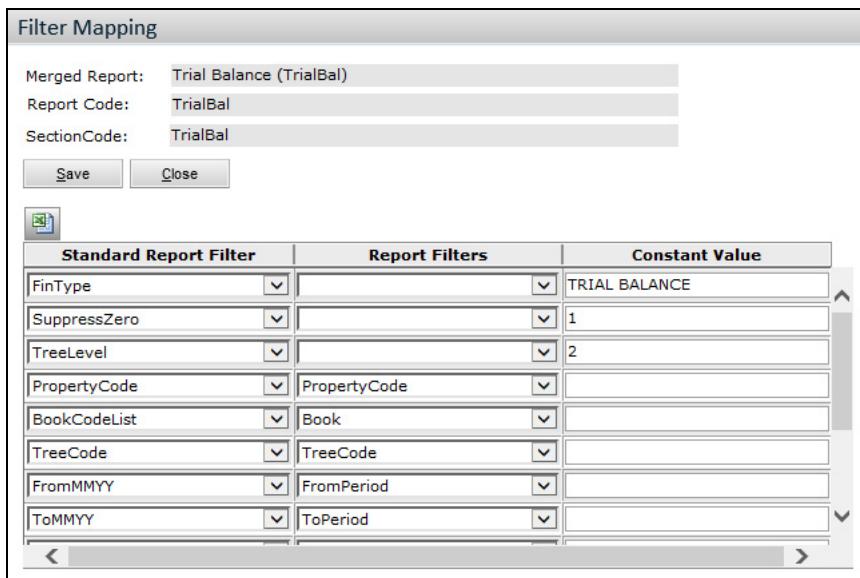
- 1 Select **Admin > YSR Correspondence > Setup Report**. The **Custom Correspondence Setup** screen appears.
- 2 Click **Submit**. The **Merged Report Setup** screen appears.

The screenshot shows the 'Merged Report Setup' dialog box. In the 'Code' field, 'TrialBal' is entered. The 'Description' field contains 'Trial Balance'. The 'Select Statement' field displays a SQL query: 'select p.hmy propertyId,p.scode sCode from property p where ...'. The 'Key Column' is set to 'propertyId'. The 'Inactive' checkbox is unchecked. On the right, the 'Notes' panel shows 'FinType : TRIAL BALANCE', 'SuppressZero : 1', 'TreeLevel : 2', and 'Filters : PropertyCode,TreeCode,FromPeriod,ToPeriod, Book, TreeCode, IsConsolidate'. Below the main setup area is a table with columns: Order, Report Code, Template File, Script File, Page Break Column, Map Filter, Sections, Inactive?, and Delete?. It contains two rows: one for 'PropHdr' with Order 5, Template File 'YSR_Prop_Header.xlsx', Script File 'YSR_property_overview.txt', Page Break Column 'propertyId', and 'Map Filter' checked; another for 'TrialBal' with Order 10, Template File 'YSR_TrialBalance.xlsx', Script File empty, Page Break Column 'propertyId', and 'Map Filter' checked.

- 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. It displays the 'Merged Report' as 'Trial Balance (TrialBal)' and the 'Report Code' as 'TrialBal'. Below are 'Save' and 'Close' buttons. The 'Report Sections' tab is selected. A table lists sections with columns: Section Code, Description, SELECT Name, Standard Report, Map Filter, Tokens, Key Columns, Primary?, and Multiple Rows?. One section is listed: 'TrialBal' with 'Trial Balance' description, 'Financial Analytics' standard report, 'Map Filter' checked, 'Tokens' checked, 'Key Columns' 'propertyId' checked, 'Primary?' checked, and 'Multiple Rows?' checked.

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



- 5 Complete the fields.
- In the **Standard Report Filter** 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 **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 Custom SQL Sub-Reports

In this section:

Named Tokens Versus Numbered Tokens	70
Named Token Example.....	72
Numbered Token Example	73
Filtering the Top-Level Select Statement.....	74
Writing WHERE Clauses	75

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 **Filter Mapping** screens. If you use numbered tokens, you must also map your filters inside the YSR report design, using the appropriate **Filter Mapping** 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 70.



YSR scripts can also contain user and session-related tokens. For more information, see "Session and User-Related Tokens" on page 38.

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 **Filter Mapping** 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								
Merged Report: Comm Charges (CommCh)								
<input type="button" value="Save"/> <input type="button" value="Close"/>								
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 **Filter Mapping** 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 Filter Mapping screen	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 Filter Mapping screen)	WHERE 1=1 #condition1# (SQL error)

Original SQL syntax	Val 1 Parameter on Filter Mapping screen	Resolved SQL syntax
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#) (incorrect syntax)	p.hmy in (#property#) (correct syntax)	WHERE 1=1 (AND p.hmy in (2)) (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 screenshot illustrates the integration of YSR filtering into a Yardi Voyager report. The 'Val 1 Parameter' window contains a SQL query that includes a placeholder for a property filter. The 'Report Filters Setup' dialog shows a configuration table where the 'Field Name' is set to 'Property', which corresponds to the placeholder in the script.

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"/>



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 ('#Date#')

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 **Filter Mapping** 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):

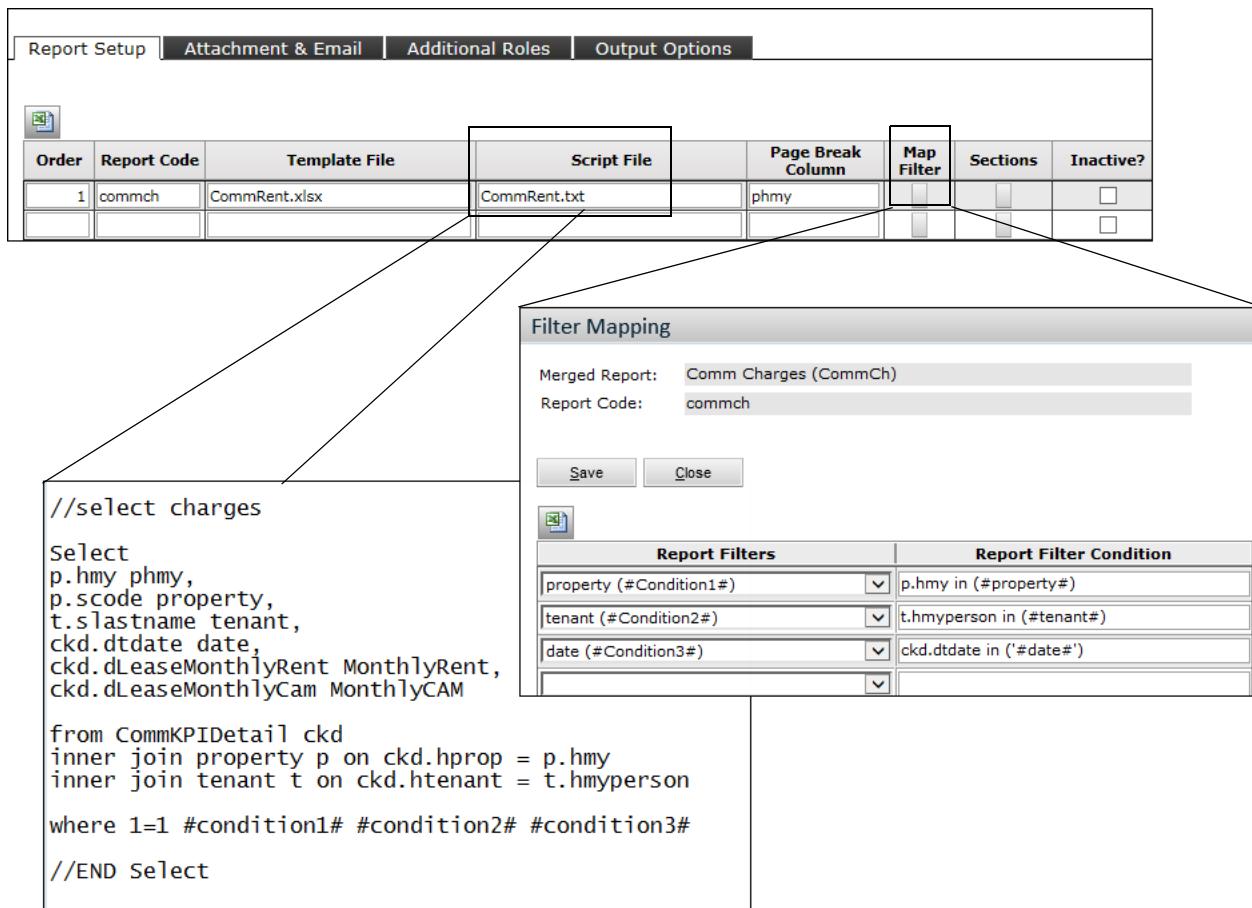
Report Filters Setup

Merged Report: Comm Charges (CommCh)

Save **Close**

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"/>	
					<input type="checkbox"/>	

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



When the user generates the report, Voyager appends the Val 1 parameters that appear on the **Filter Mapping** 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 in ('#date#')

Filtering the Top-Level Select Statement

If your report includes a top-level select statement, you must specify a filtration method for it, too. If you do not, the top-level select statement retrieves data without filtering it and expands your result set.

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:

The screenshot shows the 'Merged Report Setup' dialog box. The 'Select Statement' field contains the SQL query: `select p.scode prop, p.saddr1 name from property p where p.hmy in #property#`. The 'Key Column' is set to 'prop'. The 'Map Filters' button is highlighted.

Code	CommCh
Description	Comm Charges
Select Statement	select p.scode prop, p.saddr1 name from property p where p.hmy in #property#
Key Column	prop
Inactive	<input type="checkbox"/>
VoyagerPlus Report <input type="checkbox"/>	
Save New Define Filters Map Filters Dump SQL	

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

The screenshot shows the 'Merged Report Setup' dialog box. The 'Select Statement' field contains the SQL query: `select p.scode prop, p.saddr1 name from property p where
1=1 #condition1#`. The 'Key Column' is set to 'prop'. The 'Map Filters' button is highlighted.

A callout arrow points from the 'Map Filters' button in the Merged Report Setup dialog to the 'Filter Mapping' dialog box below.

The 'Filter Mapping' dialog box shows the following configuration:

Merged Report: Comm Charges (CommCh)							
Save Close							
	<table border="1"> <thead> <tr><th>Report Filters</th><th>Report Filter Condition</th></tr> </thead> <tbody> <tr><td>property (#Condition1#)</td><td>p.hmy in (#Property#)</td></tr> <tr><td></td><td></td></tr> </tbody> </table>	Report Filters	Report Filter Condition	property (#Condition1#)	p.hmy in (#Property#)		
Report Filters	Report Filter Condition						
property (#Condition1#)	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 194.

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
p.scode property,
t.slastname tenant,
ckd.dtdate date,
ckd.dLeaseMonthlyRent MonthlyRent,
ckd.dLeaseMonthlyCam MonthlyCAM
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.dtdate in ('#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	77
Report Section Setup	83
Report Template Setup.....	90

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

Report Section and Template Setup Overview

In this section:

Report Section Example	78
Excel Template Example.....	79
Word Template Example	80
Working with Word 2013.....	82
Report Output Example.....	83

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 and Word 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, while Word files use merge fields. 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) or merge fields (Word) 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 90.

Report Section Example

Report sections are the individual data-retrieval units that power a YSR report. 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.

The screenshot shows the 'Report Sections Setup' dialog box. At the top, it displays 'Merged Report: Credit Note Domestic (creditND)' and 'Report Code: cnd'. Below this are 'Save' and 'Close' buttons. The main area is titled 'Report Sections' and contains a table with three rows. The table columns are: Section Code, Description, SELECT Name, Standard Report, Map Filter, Tokens, Key Columns, Primary?, and Multiple Rows?. The rows are labeled 'charge', 'header', and 'total'. The 'charge' row uses a SELECT section named 'DomChargeQuery'. The 'header' and 'total' rows use a SELECT section named 'header'. The 'Primary?' column has checked boxes for 'charge' and 'header'. The 'Multiple Rows?' column has checked boxes for 'charge' and 'total'.

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

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 Report** field.

For a full definition of all the fields on the **Report Section Setup** screen, see “Setting up Report Sections” on page 84.

To review an example of an Excel template that uses these report sections, see “Excel Template Example” on page 79.

To review an example of a Word template that uses these report sections, see “Word Template Example” on page 80.

To review example report output based on these report sections, see “Report Output Example” on page 83.

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.

The diagram illustrates the structure of an Excel report template. It shows a table with two main sections: 'Header data' and 'Detail data'. The 'Header data' section contains tenant information like name, address, and contact details. The 'Detail data' section contains a breakdown of charges, including due dates, descriptions, and amounts.

CREDIT NOTE					
TENANT ATTENTION	&=Header.TenantName &=Header.BillingFirstName &=Header.BillingLastName	GST REG NO.	M9-0010744 A	INVOICE DATE	&=Header.InvoiceDate
		LEASE START	&=Header.TenantFromDate	LEASE END	&=Header.TenantToDate
		UNIT NO	&=Header.Units		
PROPERTY NAME	&=Header.PropertyName				
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

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 90.

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.

CREDIT NOTE					
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:

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

- In the last column of the table, add a merge field using the **TableEnd** keyword, followed by the section code of the expanding row data section. For example: <<TableEnd:charge>>
- Word templates use 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.

- 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
```

- 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.
- You must add a top-level select statement to any YSR report that contains a Word template.

Additional limitations of Word templates include:

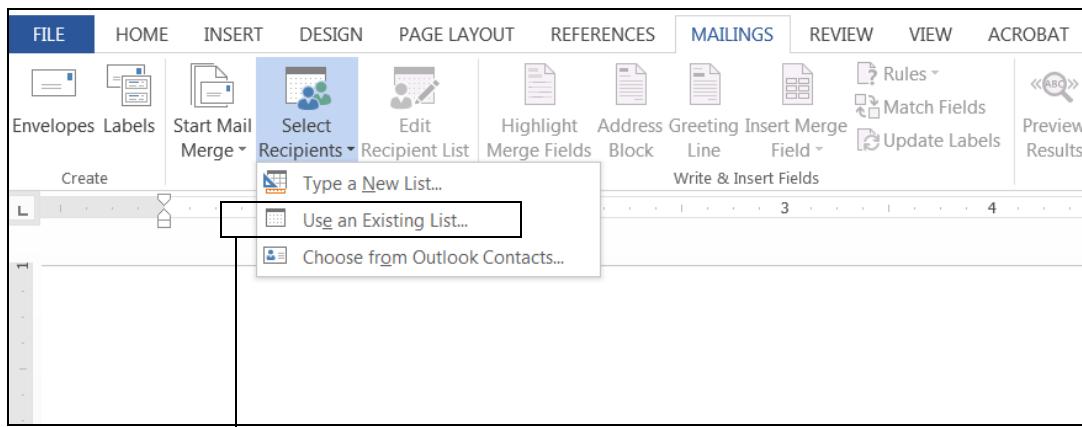
- Word does not handle tables within tables.
- Voyager does not print to screen any YSR reports that use Word templates.
- Voyager does not apply the number and date formatting of the secured user when generating Word documents.

Working with Word 2013

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

Adding Merge Fields with Word 2013

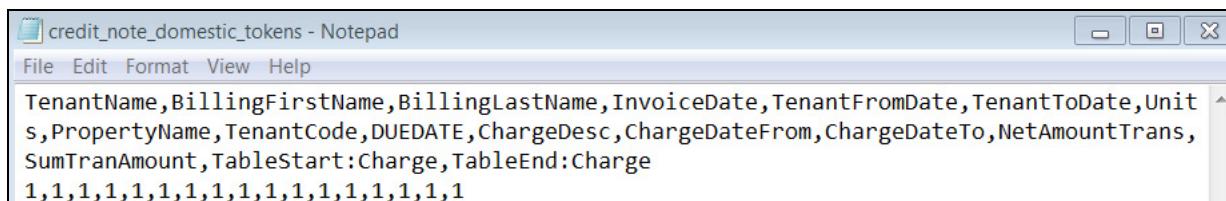
Up through Word 2010, you can add merge fields manually without identifying a data source for use in a mail merge. In Word 2013, by contrast, you must select a data source before you can insert merge fields.



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 187.

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.

Report Output Example

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

CREDIT NOTE				
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			
DUE DATE	DESCRIPTION	FROM	TO	AMOUNT
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
			Total	-1500.00

Report Section Setup

In this section:

Setting up Report Sections.....	84
Adding Static Attachments to Report Sections.....	87

This section describes how to set up report sections.

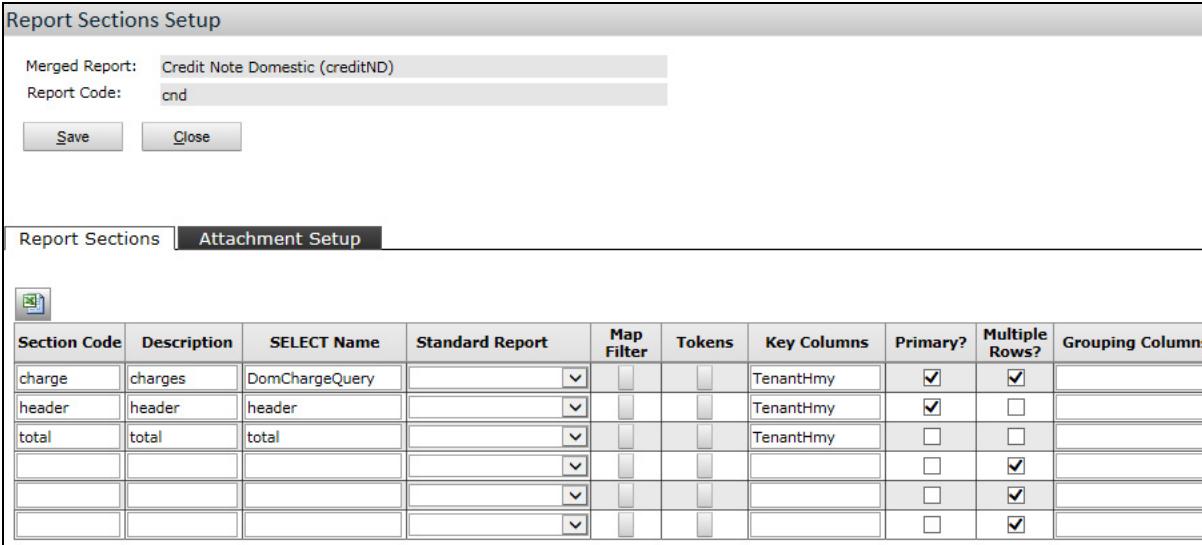
Setting up Report Sections

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

- 1 Select **Admin > YSR Correspondence > Setup Report**. The **Custom Correspondence Setup** screen appears.
- 2 Complete the filter and click **Submit**. The **Merged Report Setup** screen appears.
- 3 If it is not already active, click the **Report Setup** tab.
- 4 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 Report	Map Filter	Tokens	Key Columns	Primary?	Multiple Rows?	Grouping Columns
charge	charges	DomChargeQuery	<input type="button" value="▼"/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
header	header	header	<input type="button" value="▼"/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
total	total	total	<input type="button" value="▼"/>	<input type="checkbox"/>	<input type="checkbox"/>	TenantHmy	<input type="checkbox"/>	<input type="checkbox"/>	
			<input type="button" value="▼"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
			<input type="button" value="▼"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
			<input type="button" value="▼"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	

- 5 Complete the fields. For more information, see “Report Sections Setup Screen Reference” on page 85.
- 6 Click **Save**.
- 7 Click **Close**.

Report Sections Setup Screen Reference

Report Sections Setup

Merged Report: Credit Note Domestic (creditND)
Report Code: cnd

Save **Close**

Report Sections **Attachment Setup**

Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Key Columns	Primary?	Multiple Rows?	Grouping Columns	Summary Columns
charge	charges	DomChargeQuery	▼	□	□	TenantHmy	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
header	header	header	▼	□	□	TenantHmy	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
total	total	total	▼	□	□	TenantHmy	<input type="checkbox"/>	<input type="checkbox"/>		
			▼	□	□		<input type="checkbox"/>	<input checked="" type="checkbox"/>		
			▼	□	□		<input type="checkbox"/>	<input checked="" type="checkbox"/>		
			▼	□	□		<input type="checkbox"/>	<input checked="" 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 80.
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.
Standard 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 Filter	(Applicable to report sections that use Voyager Analytics data sources.) Opens the Filter Mapping 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 Sub-Reports” on page 27 and “Adding Voyager Analytics Sub-Reports” on page 28.
Tokens	(Applicable to report sections that use some Voyager Analytics data sources only.) Opens the Custom Token Setup screen where you can add a custom token to a Voyager analytics report. Custom tokens represent additional columns of data. NOTE The Tokens 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 sub-report” on page 28.

Key Columns	The primary key or unique identifier of the report section. Voyager uses this column to link data to other sub-report and section data.
Primary	<p>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.</p> <p>TIP If you are creating a Word template with header and detail sections, select the Primary check box for both sections.</p>
Multiple Rows	<p>(Voyager selects this check box by default for new sections.)</p> <p>Indicates that the section includes multiple rows of data.</p> <p>NOTE If Voyager returns multiple rows of data for this section but you do not select this check box, Voyager prints only the first value.</p> <p>TIP If you are configuring header data, or other data that must appear just once, clear this check box.</p>
Grouping Columns	<p>This field works in conjunction with the Summary Columns field.</p> <p>Specify a list of columns, separated by commas. Voyager returns distinct records for each column.</p> <p>NOTE Use this option only if you cannot modify your Excel template to include a Pivot Table.</p>
Summary Columns	<p>This field works in conjunction with the Grouping Columns field.</p> <p>Specify a list of columns, separated by commas. Voyager sums the column values to create distinct records for the Grouping Columns field.</p> <p>NOTE Use this option only if you cannot modify your Excel template to include a Pivot Table.</p>
Inactive	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 'Merged Report' (set to 'Lease Unpaid Invoices (XLSX0001)') and 'Report Code' (set to 'OpenInv'). Below these are 'Save' and 'Close' buttons. The bottom half of the screen is divided into two tabs: 'Report Sections' (which is the active tab, indicated by a dark brown background) and 'Attachment Setup' (indicated by a light green background). The 'Attachment Select:' section contains a multi-line text area with 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.

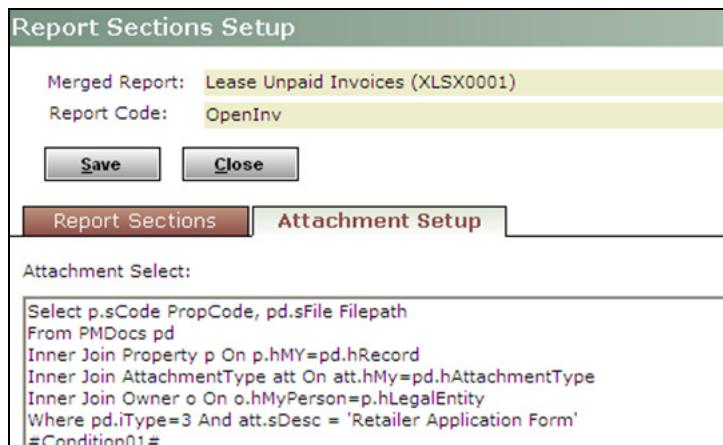
This section includes the following procedures:

To add an attachment to a sub-report	87
To add different attachments depending on user's selection.....	88
To attach an Orion document to sub-reports.....	89

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 **Merged 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 **Merged 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:

```

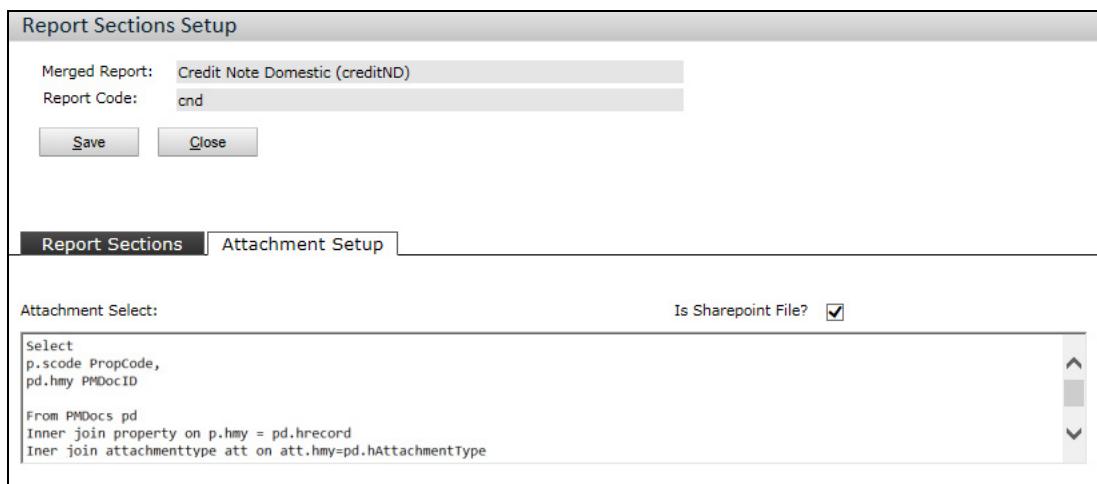
File Edit Format View Help
Select
p.sCode PropCode,
pd.hmy PMDocID

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	90
Adding Images to Word Templates	93
Displaying Text Containing HTML Tags in Word Templates.....	94
Adding an Option for Users to Select From Multiple Templates	95

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 placeholders 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 marks 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 datatable, the **Tokens** button is disabled.

To identify the field aliases used in Voyager analytics

- 1 Select **Admin > YSR Correspondence > Setup Report**. The **Custom Correspondence Setup** screen appears.
- 2 Complete the filter and click **Submit**. The **Merged 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.

The screenshot shows the 'Report Sections Setup' dialog box. At the top, it displays the 'Merged Report' as 'AP Analytics Expense Distribution (AP)' and the 'Report Code' as 'ExpenseD'. Below this are 'Save' and 'Close' buttons. The main area has tabs for 'Report Sections' and 'Attachment Setup', with 'Report Sections' currently selected. A large table lists report sections. The columns are: Section Code, Description, SELECT Name, Standard Report, Map Filter, Tokens, Key Columns, Primary?, and Multiple Rows?. The first row shows 'AP' as the Section Code, 'Expense Distribution' as the Description, and 'AP Analytics' as the Standard Report. The 'Tokens' column for this row contains a small blue square icon, indicating it's the current selection. The 'Primary?' and 'Multiple Rows?' checkboxes are checked for this row. There are three other rows in the table, each with similar columns but different data.

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

An exception has occurred.

Your request did not complete, and any changes you made have not been posted. Please try your request again.

Exception Messages:

```
SQL error : SELECT GenericQuery.* ,***** zzzzz From ( SELECT x.phmy phmy, x.tent thmy, x.trcode trcode, x.inv_date Inv_
x.duedate duedate, x.postdate PostMonth, x.goss charge_amount, case x.type when 1 then case when (x.days < 0 ) then 0 else
x.OutstandingAmount end else 0 end /* owed, */ Owed, case x.type when 1 then (case when x.days between 0 and 30 then
x.OutstandingAmount else 0 end) else 0 end [Col0], case x.type when 1 then (case when x.days between 31 and 60 then x.Outsta
else 0 end) else 0 end [Col1], case x.type when 1 then (case when x.days between 61 and 90 then x.OutstandingAmount else 0 e
end [Col2], case x.type when 1 then (case when x.days between 91 and 120 then x.OutstandingAmount else 0 end) else 0 end [C
x.type when 1 then (case when x.days >= 121 then x.OutstandingAmount else 0 end) else 0 end [Col4], (case when x.type= 2 th
x.outstandingamount else 0 end) /* Prepayments, */ Prepayments, x.ownercode, x.propcode, Isnull(x.customercode, '') customer
x.Leasecode, x.LeaseName, (case when (x.days < 0) and x.type = 1 then x.OutstandingAmount else 0 end) Future , PropAdd , Ch
Trhmy , InvoiceNo , Currency from ( Select 1 type, Isnull(o.hmyperson, 0) Ohmy , p.hmy phmy, pr.hmy tent, tr.sOtherdate4 inv_o
tr.sDateoccurred duedate, tr.upostdate postdate, N'C-' + convert(varchar,(tr.hMy-700000000)) trcode, datediff(dd, tr.sdateoccur
(m, 1, convert(datetime, N'12/31/2013', 101)) - day(convert(datetime, N'12/31/2013', 101)) days, Isnull(tr.stotalamount, 0) +
```



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.

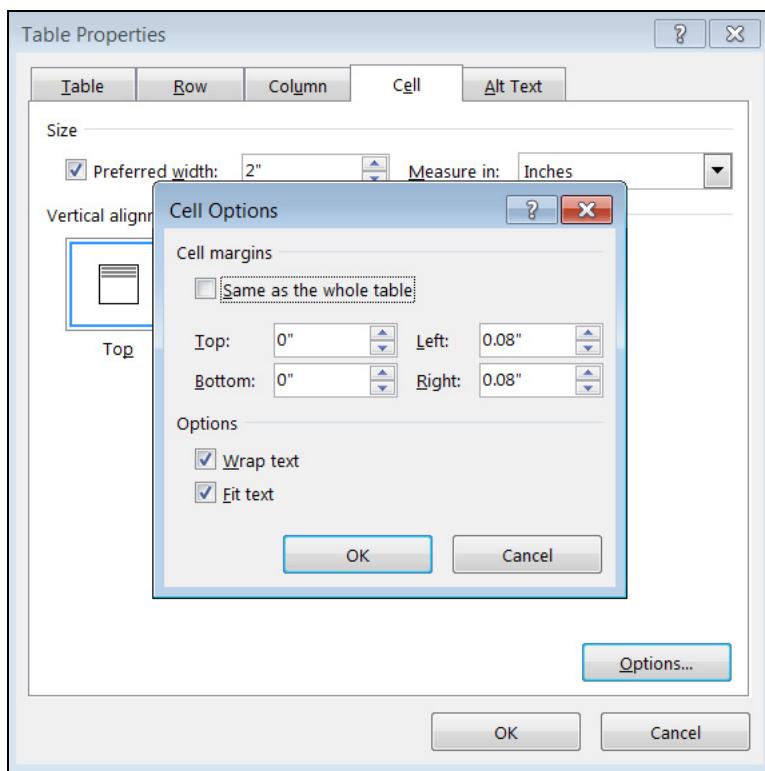
To fit contents to a cell



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	Art Design and Construction 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
from memo
where #condition1#
//End Select
```

alias with html prefix

Then use the alias in your Word template merge fields as normal:

<<htmlstext>>

YSR displays the text as formatted with HTML tags.

Adding an Option for Users to Select From Multiple Templates

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 **Merged 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)

Save **Close**

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 **Merged Report Setup** screen, enter a select statement in the **Template File** field. Use a select statement like this:

Select 'YSR_TrialBalance' + (Case '#SelCountry#' when 'de' then '_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.

Merged Report Setup

Code	TrialBal	Notes																								
Description	Trial Balance	FinType : TRIAL BALANCE SuppressZero : 1 TreeLevel : 2 Filters : PropertyCode,TreeCode,FromPeriod,ToBook, TreeCode, IsConsolidate																								
Select Statement	select p.hmy propertyId,p.scode sCode from property p where																									
	OR Script File																									
Key Column	propertyId																									
Inactive	<input type="checkbox"/>																									
<input type="button" value="Save"/> <input type="button" value="New"/>		<input type="button" value="Define Filters"/> <input type="button" value="Map Filters"/> <input type="button" value="Dump SQL"/> <input type="button" value="Dump Oracle"/> <input type="button" value="Delete Setup"/>																								
<input type="button" value="Report Setup"/> <input type="button" value="Attachment & Email"/> <input type="button" value="Additional Roles"/> <input type="button" value="Output Options"/>																										
<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 Filter</th> <th>Sections</th> </tr> </thead> <tbody> <tr> <td>10</td> <td>TrialBal</td> <td>Select 'YSR_TrialBalance' + (Case '#SelCountry#' when 'de' then '_de' when 'uk' then '_uk' else '' End) + '.xlsx'</td> <td></td> <td>propertyId</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> </tr> </tbody> </table>						Order	Report Code	Template File	Script File	Page Break Column	Map Filter	Sections	10	TrialBal	Select 'YSR_TrialBalance' + (Case '#SelCountry#' when 'de' then '_de' when 'uk' then '_uk' else '' End) + '.xlsx'		propertyId	<input type="checkbox"/>	<input type="checkbox"/>						<input type="checkbox"/>	<input type="checkbox"/>
Order	Report Code	Template File	Script File	Page Break Column	Map Filter	Sections																				
10	TrialBal	Select 'YSR_TrialBalance' + (Case '#SelCountry#' when 'de' then '_de' when 'uk' then '_uk' else '' End) + '.xlsx'		propertyId	<input type="checkbox"/>	<input type="checkbox"/>																				
					<input type="checkbox"/>	<input type="checkbox"/>																				

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.

CHAPTER 5

Generating YSR Reports

In this chapter:

Generating YSR Reports.....	98
Task Runner	100
Report Scheduler.....	104
Custom Side Menus for YSR Reports in Voyager	106
Custom Side Menus for YSR Reports in VoyagerPlus	108

This section describes options for generating YSR reports.

Generating YSR Reports

When you generate reports with YSR, you have many choices about how to view your results. Your primary choice is whether to generate a YSR report directly to screen or to generate reports in PDF or Excel.

Administrators commonly design YSR reports to generate multiple reports at once, in which case it can be unwieldy to display all results directly on screen. Instead, select **PDF** or **Excel** in the **Output Type** field when generating reports. YSR generates as many reports as demanded by the filter criteria, as distilled by the top-level select statement, and displays a list with links to each report. For example, the next graphic shows a custom report for use with Investment Management. This report generates transaction history for every investor in a fund.

1) Select the YSR report you want to generate and determine how you want to view results.

2) Enter filter criteria.

investorhmy	investorcode	investorname	Report
143.00	yinv1	Investor1	View Report
144.00	yinv2	Investor2	View Report
145.00	yinv3	Investor3	View Report
146.00	yinv4	Investor4	View Report
147.00	yinv5	Investor5	View Report

3) Depending on how the report is set up, Voyager may display a list with links to each report.

In some cases, however, you may want to use YSR to review a single report on screen. If you choose a report that uses an Excel template and select **Screen** in the **Output Type** field, Voyager generates a report on screen. For example, here is a Commercial Rent Roll 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



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 also email reports as attachments, publish to SharePoint, or show them on Portal. Your administrator controls permissions for attaching and emailing reports.

To generate a YSR report

- 1 Select **Admin > YSR Correspondence > Generate Report**. The report-generation screen appears.
- 2 Select the name of the YSR report you want to generate from the **Report Name** field. The custom filter fields associated with the YSR report appear.
- 3 Complete the fields.
- 4 Click **Generate**. Voyager generates the report.

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.ysiYSRCorrespondenceTask`

Example URL

`ReportCode=ClientST&OutputType=PDF&Merge=Yes&Attach=Yes&Email=Yes&Generate=Yes&Grid=Yes&OwnerId=glegal&AcctID=111-000-00&DateFrom=01/2012&DateTo=12/2013`



To set up the task, you must enter the URL on the **Task Runner Step** screen. Alternatively, you can save the URL in a text file in your Reports path and enter, in lieu of the URL, the following: `URLTemplate=filename.txt` (substitute the filename of your text file).

For an example, see “To generate YSR reports using Task Runner” on page 103.

URL Components

Parameter	Parameter Value	Mandatory	Example
<code>ReportCode=</code>	The code of the report you want to generate.	Yes	<code>ClientST</code>

Parameter	Parameter Value	Mandatory	Example
&OutputType=	PDF, Excel, or Screen. The format in which Voyager generates the report.		PDF
&Attach=	Yes or No. If &Attach=Yes, Voyager sends the report as an attachment.		Yes
&Email=	Yes or No. If &Email=Yes, Voyager emails the report to recipients.		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
&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 Report Filters Setup screen of the YSR report, ClientST.	As specified on the Report Filters Setup 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 Report Filters Setup screen of the YSR report, ClientST.	As specified on the Report Filters Setup screen.	111-000-00
&DateFrom=	In this example, it is necessary to include the DateFrom parameter because this particular report (ClientST) has a mandatory date range filter. "DateFrom" is the field name specified on the Report Filters Setup screen of the YSR report, ClientST.	As specified on the Report Filters Setup screen.	01/2012
&DateTo=	In this example, it is necessary to include the DateTo parameter because this particular report (ClientST) has a mandatory date range filter. "DateTo" is the field name specified on the Report Filters Setup screen of the YSR report, ClientST.	As specified on the Report Filters Setup screen.	12/2013

The custom URL detailed in the preceding material generates this screen:

Client Statement	
Owner	glegal
Property	
Account	111-000-00
From	01/2012
To	12/2013
Output Type: PDF Merge Reports: <input checked="" type="checkbox"/> Show Grid: <input checked="" type="checkbox"/> <input type="button" value="Generate"/> <input type="button" value="Clear"/>	
Attach Reports: <input checked="" type="checkbox"/> Email Reports: <input checked="" type="checkbox"/>	

As illustrated in the preceding graphic, Voyager uses the parameter values specified in the custom URL to select reporting options and to define the values in the custom filter fields on the left side of the screen.

Because every YSR report has its own custom filter fields, each custom URL must contain parameter values that correspond to the custom filter fields. For example, when adding a URL to execute the Client Statement report, you must add a parameter for the **Owner**, **Account**, **From**, and **To** fields. The name of each parameter is the field name that you give to each filter on the **Report Filters Setup** screen.

In this example, the parameter names for the mandatory filter fields are **OwnerId**, **AcctID**, **FromDate**, and **ToDate**.

The screenshot shows the 'Report Filters Setup' screen for a 'Client Statement' merged report. The interface includes a toolbar with 'Save' and 'Close' buttons, and a table for defining filter fields. The table columns are: Field Name, Label, Sequence, Type, Lookup Name, Multi Select?, Parent, Mandatory?, and Link. The rows represent the five filter fields defined for this report:

Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	Link
OwnerId	Owner	01	Lookup List	ysiOwnerLookup	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
PropertyID	Property	02	Lookup List	ysiPropertyOrListLookup	<input checked="" type="checkbox"/>	01	<input type="checkbox"/>	
AcctID	Account	03	Lookup List	ysiAccountCashLookup	<input type="checkbox"/>		<input checked="" type="checkbox"/>	
FromDate	From	04	Post Month		<input type="checkbox"/>		<input checked="" type="checkbox"/>	
ToDate	To	05	Post Month		<input type="checkbox"/>		<input checked="" type="checkbox"/>	
					<input type="checkbox"/>		<input type="checkbox"/>	
					<input type="checkbox"/>		<input type="checkbox"/>	

The field names specified on the **Report Filters Setup** screen become the parameters that you can include in the custom URL you set up when adding a Task Runner task.



At minimum, you must include in your Task Runner URL the parameters (and their values) for all the mandatory filter fields in the YSR report.



For multi-select filter fields, you can pass in multiple values by separating them with a caret (^). For example &OwnerId=glegal^hlegal instructs Task Runner to generate the YSR report for two owners.

To generate YSR reports using Task Runner



At minimum, you must include in your Task Runner URL the parameters (and their values) for all the mandatory filter fields in the YSR report.

- 1 Select **Admin > Task Runner > Task > Add**. The **Task Runner Task** screen appears.

The screenshot shows the 'Task Runner Task' configuration screen. It includes fields for 'Code' (YSR), 'Name' (YSR reports), and 'Description' (YSR monthly report packet). At the bottom are buttons for 'Save', 'New', 'Delete', and '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.

The screenshot shows the 'Task Runner Step' configuration screen. It includes fields for 'Template' (APPTASK), 'Name' (Application Task d57c260b-8b43-402b-91d4-e6a5bfe5b7de), and 'Description' (Executes a YSI.NET class). Below these are 'Order' (0), 'Inactive' (unchecked), and a checkbox for 'Execute only if task already failed'. At the bottom are buttons for 'Save', 'New', 'Delete', and 'Help'. A 'Details' tab is open, showing a table with columns 'Name' and 'Value' containing entries for 'ClassName', 'PropertySecurityUser', and '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

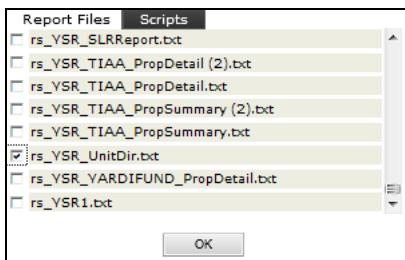
You can use Voyager Report Scheduler to generate YSR reports automatically at a designated time. To generate YSR reports via Report Scheduler, you must:

- 1 Make the YSR report available to Report Scheduler.
- 2 Complete the standard setup procedures for Report Scheduler.

This section describes step one. For more information about step two, see the *Service Manager Administration and Setup Guide*.

Making YSR Reports Available to Report Scheduler

In order to schedule YSR reports through Voyager Report Scheduler, you must first make them available to the Report Scheduler. You can make a report available to Report Scheduler by creating a script file that adds the YSR report to the Report Scheduler database and defines the filter fields associated with the YSR report. After you save the file to the Reports path with the correct naming conventions, the file becomes accessible via Report Scheduler. For example, the following graphic shows several YSR report files available for scheduling in Report Scheduler.



The example file, rs_YSR_UnitDir.txt, contains the following information:

```
//Vista

//Database
analytic YSI.Utils.CustomCorrespondence#YSI.Utils.CustomCorrespondence.ySiYSRConductorInterface
//End Database

//Title
YSR Unit Dir
//End Title

//Select
Select distinct sCode from property p where 1=1
#CONDITION#

//End Select

//Columns
//Type Name Head1 Head2 Head3 Head4 Show Color Formula Drill Key Width
T, , , , , Property, Y, , , 3, , 500,
//End Columns

//Filter
//Type, DataType, Name Caption, Key, List, Val1, Val2, Mandatory, Mu
C, T, PropertyId, PropertyId, N, 524, p.hmy in (#PropertyId#), , Y, Y,
C, T, ReportCode, ReportCode, N, "Select 'UnitDir'", , , Y, Y, N
//End Filter
```

As in the example, the text file must meet the following criteria:

- It must define the YSR Conductor Interface class name in the //Database section, exactly as illustrated in the preceding graphic:
analytic YSI.Utils.CustomCorrespondence#YSI.Utils.CustomCorrespondence,ySiYSRConductorInterface
- In the //FILTER section, it must define the filters that the user completes when scheduling the report, such that the named filter elements exactly match the filter names in YSR and all mandatory filter fields are included.
- An additional filter field must contain the value “Select ‘ReportCode’”, with its list set to “Select ‘The YSR Report Code’”
- The filename must begin with rs_YSR_.

After you create the Report Scheduler file and save it to the Reports path, the YSR report is available for scheduling in Report Scheduler.



You can also use the Yardi Excel Add-In to create the script that adds your YSR report to the Report Scheduler database. When using the Yardi Excel Add-In to build a report (Dump SQL), select the option to prepare the report for Report Scheduler. For more information, see “Dump SQL Tab Screen Reference” on page 197.

Custom Side Menus for YSR Reports in Voyager

You can create a custom side menu that directs users to one or more specific YSR reports that you have already set up. This enables your users to take advantage of YSR reporting without having to complete setup tasks.

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

```
./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.	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.		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

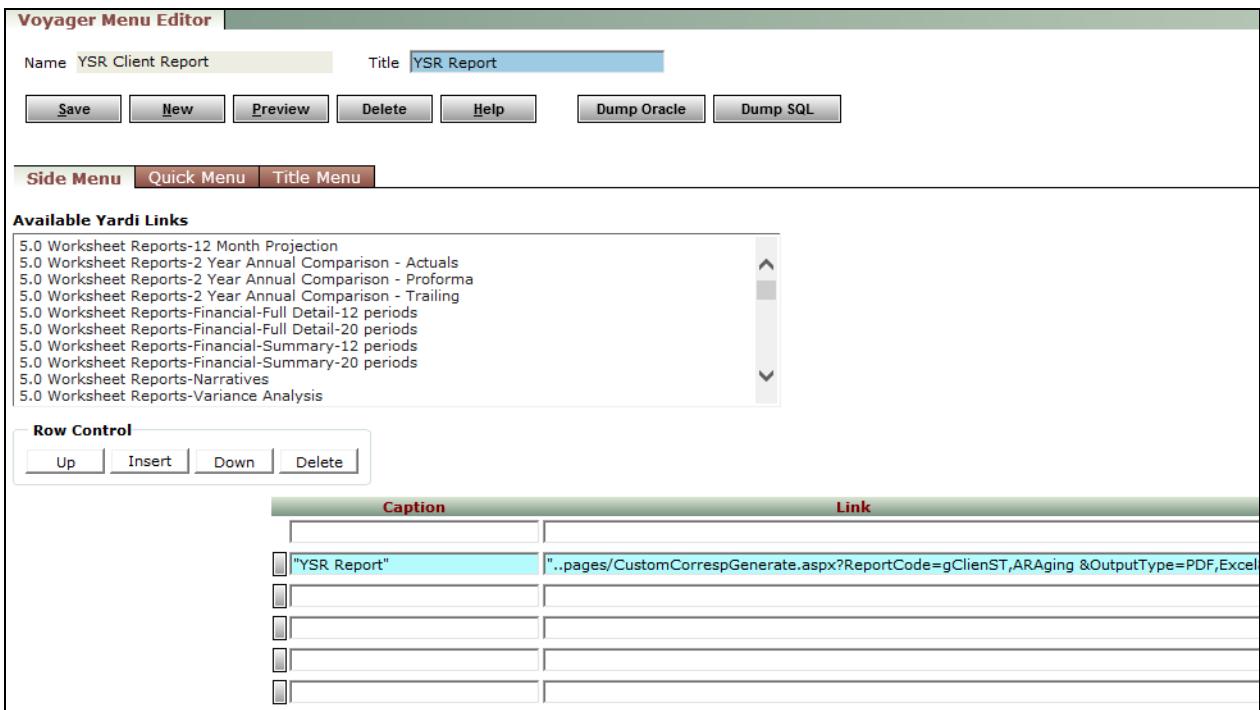
The custom URL detailed in the preceding sections creates a YSR report-generation screen. On this screen, users can choose between a Client Statement report (ClientST) and an AR Aging report (AgingSum), as depicted here:

The screenshot shows a user interface for generating YSR reports. At the top, there are dropdown menus for 'Report Name' containing 'Client Statement (ClientST)' and 'AR Aging Summary Report (AgingSum)', and 'Output Type' containing 'PDF'. Below these are several checkboxes: 'Merge Reports' (checked), 'Email Reports' (unchecked), 'Show Grid' (checked), and 'Print Preview' (unchecked). At the bottom are 'Generate' and 'Clear' buttons.

To add a custom side menu for YSR reports

- From Voyager System Administration, select **Security > Add Menus**. The **Create New Menu** screen appears.

- 2** Complete the required fields and click **OK**. The **Voyager Menu Editor** screen appears.



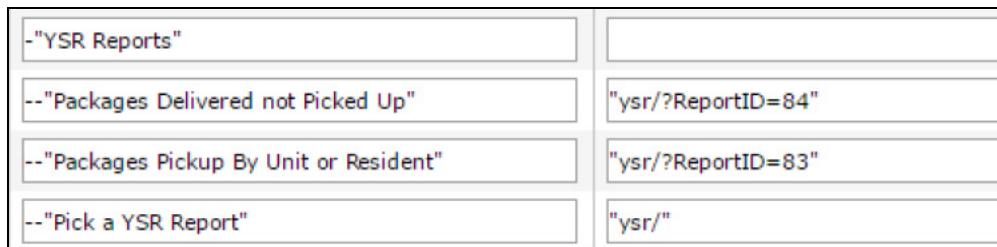
- 3** Add a caption and enter the custom URL for your YSR report in the **Link** field.

- 4** Click **Save**.

Custom Side Menus 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).



To create the links to specific YSR reports, you must know the ReportID of your YSR report. 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

CHAPTER 6

Custom Financial Analytics Reports in YSR

In this chapter:

CFA Report Setup Overview.....	112
CFA Account Tree Reports in YSR	115
CFA Property Portfolio Reports in YSR.....	126
CFA Comparison Reports in YSR	132

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.....	112
Working with Consolidated Data	113
Formatting CFA Reports.....	114

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 with the 'Report Sections' tab selected. It displays a table for defining report sections. A modal dialog, 'Filter Mapping', is open over the main window. This modal has its own 'Report Sections' tab, which is also selected. It contains fields for 'Merged Report' (set to 'CFA Balance Sheet (cfabs)'), 'Report Code' (set to 'cfabs'), and 'SectionCode' (set to 'CF'). Below these are 'Save' and 'Close' buttons. At the bottom of the main window, another modal dialog, 'Standard Report Filter', is open. It has three tabs: 'Standard Report Filter', 'Report Filters', and 'Constant Value'. The 'Standard Report Filter' tab is active, showing a table with rows for 'Template' (set to 'CFA_BS'), 'PropertyCode', 'FromMMYY', 'ToMMYY', and 'SuppressZero'. The 'Report Filters' and 'Constant Value' tabs are also visible but not active.

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]

Working with Consolidated Data

If you want your report 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	
Property	comoff01^comind01
From Date	01/2007
To Date	12/2007
Consolidate	<input checked="" type="checkbox"/>
Report Name	CFA Balance Sheet (cfabs)
Output Type	Screen
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"/>	

The **Consolidate** check box is mapped to the **IsConsolidate** filter element, as depicted below.

The screenshot shows two overlapping windows. The top window is titled "Report Sections Setup" and contains fields for "Merged Report: CFA Balance Sheet (cfabs)" and "Report Code: cfabs", with "Save" and "Close" buttons. The bottom window is titled "Filter Mapping" and contains fields for "Merged Report: CFA Balance Sheet (cfabs)", "Report Code: cfabs", and "SectionCode: CF", also with "Save" and "Close" buttons. Both windows have tabs for "Report Sections" and "Attachment Setup". Below these windows is a larger table titled "Standard Report Filter" with columns for "Standard Report Filter", "Report Filters", and "Constant Value". The "Report Filters" column contains dropdowns for "Template" (set to "CFA_BS"), "PropertyCode" (set to "Property"), and "IsConsolidate" (set to "Consolidate").

Standard Report Filter	Report Filters	Constant Value
Template		CFA_BS
PropertyCode	Property	
IsConsolidate	Consolidate	

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 116.

CFA formatting field names and values

Field Name	Values	Supported in YSR
bBold	TRUE	YES
	FALSE	
iUnderline	0 (no underlining)	YES
	1 (single underline)	
	2 (double underline)	
iIndent	TRUE	NO
	FALSE	
bItalic	TRUE	YES
	FALSE	
bNegate	TRUE	NO
	FALSE	
iType	TRUE	NO
	FALSE	
iAfter	TRUE	NO
	FALSE	
iNest	TRUE	NO
	FALSE	

CFA Account Tree Reports in YSR

In this section:

Account Tree Report with Bold and Underline	116
Account Tree Report with Manipulated Data.....	121

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

- 1 Review your CFA report in Voyager.

Custom Financial Reports

Property: comoff01
Book: Accrual
Report Template: CFA_BS
Period: 12/2007 to 12/2007
Summary By: Client
Account Tree: mod_bs

Denominator:
Department:
Source:
Immediate Source:
DecimalDigits: 0 **SUPPRESS ZERO**
Show Account: **Show Column:** **Detail:**
Summary: **Tree Level:** 1 **Grid:**

	Value
1000-0000 ASSETS	
1001-0000 CURRENT ASSETS	
1100-0000 CASH & CASH EQUIVALENT	
1110-0000 Cash - Operating	482,208
1190-0000 TOTAL CASH & CASH EQUIVALENT	482,208
1499-0000 TOTAL CURRENT ASSETS	482,208
1500-0000 FIXED ASSETS	
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

- 2 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:**

Columns: 1 **Apply User Group Security:**

Buttons: Edit, New, Close, Help, Delete, Save As, Set Permissions

Financial Formulas:

New Row: **Function:** Copy Columns **From:** Column 1 **To:** Column 1 **Execute:**

ID	From Acct	To Acct	Column 1	Delete
1			Value	<input type="button"/>
2			MTD	<input type="button"/>

- 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	Account Name	Account Code	Value	bBold	iUnderline
4	&=[CF].[acctDesc]	&=[CF].[acctcode]	&=[CF].[column1]	&=[CF].[bBold]	&=[CF].[iUnderline]
5					

- b Use conditional formatting to apply bold formatting when column D has a value of TRUE.

A	B	C	D	E	F	G
1						
2						
3	Account Name	Account Code	Value	bBold	iUnderline	
4	&=[CF].[acctDesc]	&=[CF].[acctcode]	&=[CF].[column1]	&=[CF].[bBold]	&=[CF].[iUnderline]	
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						



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.

Merged Report Setup

Code	cfabs	Notes
Description	CFA Balance Sheet	
Select Statement		
OR Script File		
Key Column		
Inactive	<input type="checkbox"/>	VoyagerPlus Report <input type="checkbox"/>
<input type="button" value="Save"/> <input type="button" value="New"/> <input type="button" value="Define Filters"/> <input type="button" value="Map Filters"/> <input type="button" value="Dump SQL"/> <input type="button" value="Dump Oracle"/> <input type="button" value="Delete Setup"/>		
<input type="button" value="Report Setup"/> <input type="button" value="Attachment & Email"/> <input type="button" value="Additional Roles"/> <input type="button" value="Output Options"/>		

Report Setup

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

5 Add custom filter fields to your YSR report.

Report Filters Setup

Merged Report: CFA Balance Sheet (cfabs)

<input type="button" value="Save"/>	<input type="button" value="Close"/>																																							
<table border="1"> <thead> <tr> <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> </tr> </thead> <tbody> <tr> <td>Property</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> </tr> <tr> <td>FromDate</td> <td>From Date</td> <td>2</td> <td>Post Month</td> <td></td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td>ToDate</td> <td>To Date</td> <td>3</td> <td>Post Month</td> <td></td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> </tr> </tbody> </table>									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"/>
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.

Report Sections Setup

Merged Report: CFA Balance Sheet (cfabs)

Report Code: cfabs

<input type="button" value="Save"/>	<input type="button" value="Close"/>																																		
<input type="button" value="Report Sections"/> <input type="button" value="Attachment Setup"/>																																			
<table border="1"> <thead> <tr> <th>Section Code</th> <th>Description</th> <th>SELECT Name</th> <th>Standard Report</th> <th>Map Filter</th> <th>Tokens</th> <th>Key Columns</th> <th>Primary?</th> <th>Multiple Rows?</th> </tr> </thead> <tbody> <tr> <td>CF</td> <td>Custom Financial Analytics</td> <td></td> <td>Custom Financials</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> </tbody> </table>									Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Key Columns	Primary?	Multiple Rows?	CF	Custom Financial Analytics		Custom Financials	<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"/>
Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Key Columns	Primary?	Multiple Rows?																											
CF	Custom Financial Analytics		Custom Financials	<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"/>																											

- 7 Map filters. Map the **Template** element in the **Standard Report Filter** column to the code of the CFA report you want to generate in YSR.

Filter Mapping

Merged Report:	CFA Balance Sheet (cfabs)	
Report Code:	cfabs	
SectionCode:	CF	
<input type="button" value="Save"/> <input type="button" value="Close"/>		
 <input type="button" value="Edit"/>		
Standard Report Filter	Report Filters	Constant Value
Template		CFA_BS
PropertyCode	Property	
FromMMYY	FromDate	
ToMMYY	ToDate	
SuppressZero		True



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 Report Filter** 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"/>	
Account Name	Account Code	Value	
ASSETS	1000-0000		
CURRENT ASSETS	1001-0000		
CASH & CASH EQUIVALENT	1100-0000		
Cash - Operating	1110-0000	482,208.37	
TOTAL CASH & CASH EQUIVALENT	1190-0000	482,208.37	
TOTAL CURRENT ASSETS	1499-0000	482,208.37	
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	
TOTAL ACCUMULATED DEPRECIATION	1690-0000	-51,302.00	

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 Total Revenue	Sum of Total Rent and Total Recoverables	
2 \$1,444,787.67	\$1,450,521.11	

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	
Book	Accrual	Assistant Manager	
Report Template	CFAIS	Client	
Period	12/2007	Country	
Summary By	Client	District Manager	
Account Tree	is_ysr	Fund Objective	
		Freeze Pane	<input type="checkbox"/>
			<input checked="" type="checkbox"/> Show Account
			<input checked="" type="checkbox"/> Summary
			<input checked="" type="checkbox"/> DecimalDigits
			<input checked="" type="checkbox"/> Immediate S
			<input checked="" type="checkbox"/> Source
			<input checked="" type="checkbox"/> Department
			<input checked="" type="checkbox"/> Denominator
			<input checked="" type="checkbox"/> Up
			<input checked="" type="checkbox"/> Down
			<input checked="" type="checkbox"/> Left
			<input checked="" type="checkbox"/> Right
		Extracted Data	
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	
	TOTAL COMMON RECOVERABLE	51,864	
6100-0000	REAL ESTATE TAXES - RECOVERABLE		
6110-0000	Real Estate Taxes	89,106	

2 Review your CFA report design.

ID	From Acct	To Acct	Column 1	Delete
1			Extracted D...	
2	41100000	45250000	QTD	
3	51000000	54100000	QTD	
4	80600000	85100000	QTD	
5	60150000	69500000	QTD	
6	85200000	85500000	QTD	
7	75100000	76250000	QTD	

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.

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.

- b** On the second worksheet, use Excel functions to pick out data and manipulate it.

	A	B	C	D
1	Value Column	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.

- 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
Total Revenue	Sum of Total Rent and Total Recoverables
=CFTotalRev	=CFRentRecov

- e** Save your Excel template to the Reports path.

5 Add a YSR report.

Merged Report Setup

Code	cfaus	Notes																								
Description	CFA Inc Statement manipulation																									
Select Statement																										
OR Script File																										
Key Column																										
Inactive	<input type="checkbox"/>	VoyagerPlus Report <input type="checkbox"/>																								
Save New		Define Filters Map Filters Dump SQL Dump Oracle Delete Setup																								
Report Setup Attachment & Email Additional Roles Output Options																										
<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 Filter</th> <th>Sections</th> <th>Inactive?</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>cfaus</td> <td>cfa_is_template.xlsx</td> <td></td> <td></td> <td></td> <td></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> </tr> </tbody> </table>			Order	Report Code	Template File	Script File	Page Break Column	Map Filter	Sections	Inactive?	1	cfaus	cfa_is_template.xlsx					<input type="checkbox"/>								<input type="checkbox"/>
Order	Report Code	Template File	Script File	Page Break Column	Map Filter	Sections	Inactive?																			
1	cfaus	cfa_is_template.xlsx					<input type="checkbox"/>																			
							<input type="checkbox"/>																			

6 Add custom filter fields to your YSR report.

Report Filters Setup

Merged Report: CFA Inc Statement manipulation (cfaus)																																																								
<input type="button" value="Save"/> <input type="button" value="Close"/>																																																								
<table border="1"> <thead> <tr> <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> </tr> </thead> <tbody> <tr> <td>Property</td> <td>Property</td> <td>1</td> <td>Lookup List</td> <td>ysiPropertyLookup</td> <td><input type="checkbox"/></td> <td></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>FromDate</td> <td>From Date</td> <td>2</td> <td>Post Month</td> <td></td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td>ToDate</td> <td>To Date</td> <td>3</td> <td>Post Month</td> <td></td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> </tr> </tbody> </table>									Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?	Property	Property	1	Lookup List	ysiPropertyLookup	<input 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"/>						<input type="checkbox"/>		<input type="checkbox"/>						<input type="checkbox"/>		<input type="checkbox"/>
Field Name	Label	Sequence	Type	Lookup Name	Multi Select?	Parent	Mandatory?																																																	
Property	Property	1	Lookup List	ysiPropertyLookup	<input 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"/>																																																	
					<input type="checkbox"/>		<input type="checkbox"/>																																																	
					<input type="checkbox"/>		<input type="checkbox"/>																																																	

7 Select Custom Financials as your data source.

Report Sections Setup

Merged Report: CFA Balance Sheet (cfabs)																																			
Report Code: cfabs																																			
<input type="button" value="Save"/> <input type="button" value="Close"/>																																			
Report Sections Attachment Setup																																			
<table border="1"> <thead> <tr> <th>Section Code</th> <th>Description</th> <th>SELECT Name</th> <th>Standard Report</th> <th>Map Filter</th> <th>Tokens</th> <th>Key Columns</th> <th>Primary?</th> <th>Multiple Rows?</th> </tr> </thead> <tbody> <tr> <td>CF</td> <td>Custom Financial Analytics</td> <td></td> <td>Custom Financials</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> </tbody> </table>									Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Key Columns	Primary?	Multiple Rows?	CF	Custom Financial Analytics		Custom Financials	<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"/>
Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Key Columns	Primary?	Multiple Rows?																											
CF	Custom Financial Analytics		Custom Financials	<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"/>																											

- 8** Map filters. Map the **Template** element in the **Standard Report Filter** column to the code of the CFA report you want to generate in YSR. Enter the CFA report code in the **Constant Value** column.

Filter Mapping

Merged 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="Edit"/> <table border="1"> <thead> <tr> <th>Standard Report Filter</th> <th>Report Filters</th> <th>Constant Value</th> </tr> </thead> <tbody> <tr> <td>Template</td> <td></td> <td>cfais</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></td> <td></td> <td></td> </tr> </tbody> </table>			Standard Report Filter	Report Filters	Constant Value	Template		cfais	PropertyCode	Property		FromMMYY	FromDate		ToMMYY	ToDate				
Standard Report Filter	Report Filters	Constant Value																		
Template		cfais																		
PropertyCode	Property																			
FromMMYY	FromDate																			
ToMMYY	ToDate																			



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 Report Filter** 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	127
Property Portfolio Report, Summarized by Property.....	131

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
230,450	1,780	228,670	99
189,650	31,000	158,650	84
189,650	31,000	158,650	84
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' (ysl_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, a 'Financial Formulas' tab is open, showing 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: Row 1 shows 'Total Rent' in Column 1, 'Vacancy' in Column 2, 'Rent Less V...' in Column 3, and 'EO%' in Column 4; Row 2 shows 'MTD' in Column 1, 'MTD' in Column 2, 'Formula' in Column 3, and 'Formula' in Column 4. A 'Delete' button is also present in the table header.

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, B, C, D, and E. Row 1 contains a single cell with the value '1'. Row 2 contains a cell with the value 'Economic Occupancy by Market'. Row 3 is empty. Row 4 contains cells with the values 'Summarize By', 'Total Rent', 'Vacancy', 'Rent Less Vac', and 'EO %'. Row 5 contains cells with the formulas '&=[CF].[acctDesc]', '&=[CF].[column1]', '&=[CF].[column2]', '&=[CF].[column3]', and '&=[CF].[column4]'. Row 6 is empty.

To set up a property portfolio report, summarized by attribute

- 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
Freeze Pane <input type="checkbox"/>					
	Total Rent	Vacancy	Rent Less Vac	EO%	
Raleigh/Durham					
Roosevelt Tower	230,450	1,780	228,670	99	
Total Raleigh/Durham	230,450	1,780	228,670	99	
Phoenix					
Sunrise Tower	189,650	31,000	158,650	84	
Total Phoenix	189,650	31,000	158,650	84	
Grand Total	420,100	32,780	387,320	92	

- Add a YSR report.

Merged Report Setup

Code	cfa_eo_by_market	Notes																																			
Description	CFA Economic Occupancy by Market Attribute																																				
Select Statement																																					
Key Column																																					
Inactive	<input type="checkbox"/>	VoyagerPlus Report <input type="checkbox"/>																																			
Save New		Define Filters Map Filters Dump SQL Dump Oracle Delete Setup																																			
<table border="1"> <tr> <td>Report Setup</td> <td>Attachment & Email</td> <td>Additional Roles</td> <td>Output Options</td> </tr> <tr> <td colspan="4"> <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 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>CFheader.txt</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> </td> </tr> </table>			Report Setup	Attachment & Email	Additional Roles	Output Options	<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 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>CFheader.txt</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 Filter	Sections	Inactive?	Delete?	1	eo	ysr_cfa_eobymarket.xlsx	CFheader.txt				<input type="checkbox"/>	<input type="checkbox"/>								<input type="checkbox"/>	<input type="checkbox"/>
Report Setup	Attachment & Email	Additional Roles	Output Options																																		
<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 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>CFheader.txt</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 Filter	Sections	Inactive?	Delete?	1	eo	ysr_cfa_eobymarket.xlsx	CFheader.txt				<input type="checkbox"/>	<input type="checkbox"/>								<input type="checkbox"/>	<input type="checkbox"/>							
Order	Report Code	Template File	Script File	Page Break Column	Map Filter	Sections	Inactive?	Delete?																													
1	eo	ysr_cfa_eobymarket.xlsx	CFheader.txt				<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 **Close**

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"/>		
					<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

Merged Report: CFA Economic Occupancy by Market Attribute (cfa_eo_by_market)

Report Code: eo

Save **Close**

Report Sections **Attachment Setup**

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

5 Map filters.

Filter Mapping

Merged Report:	CFA Economic Occupancy by Market Attribute (cfa_eo_by_market)	
Report Code:	eo	
SectionCode:	CF	
<input type="button" value="Save"/> <input type="button" value="Close"/>		
		
Standard Report Filter	Report Filters	Constant Value
Template	Property	eo
PropertyCode	PostMonth	
FromMMYY	PostMonth	
ToMMYY	AttributeSortName	
AttributeSortName	AttributeSortName	
AttributeSortValue	AttributeSortName	
SuppressZero	SuppressZero	
BookCodeList	Book	
Detail	Detail	



Map the **Book** filter field to the **BookCodeList** element in the **Standard Report Filter** 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

<u>Property</u> comoff01^comoff02	<u>CF Type</u> Portfolio by Attribute	<u>Post Month</u> 01/2017	<u>Report Name</u> CFA Economic Occupancy by Market Attribute (<input type="button" value="Screen"/>	<input type="checkbox"/> Attach Reports
<u>Book</u>		<input checked="" type="checkbox"/> Suppress Zero	<input type="checkbox"/> Merge Reports	<input type="checkbox"/> Email Reports	<input type="checkbox"/> Publish To SharePoint
<input type="checkbox"/> Summarize by Market		<input type="checkbox"/> Show Grid		<input type="checkbox"/> Show on Portal	
<u>Detail</u> 0		<input type="button" value="Generate"/>	<input type="button" value="Clear"/>		

Economic Occupancy by Market

Summarize By	Total Rent	Vacancy	Rent Less Vac	EO %
Market				
Raleigh/Durham	230450	1780	228670	99.22759818
Phoenix	189650	31000	158650	83.65409966
Total Market	420100	32780	387320	92.19709593
Grand Total	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 **Close**

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

Filter Mapping

Merged Report: CFA Economic Occupancy by Property (cfa_eo_by_prop)

Report Code: 1

SectionCode: CF

Save **Close**

Standard Report Filter	Report Filters	Constant Value
Template		eo
PropertyCode	Property	
FromMMYY	PostMonth	
ToMMYY	PostMonth	
BookCodeList	Book	
Detail		1
SuppressZero	SuppressZero	

CFA Comparison Reports in YSR

In this section:

Property Comparison Reports in YSR	132
Attribute Comparison Reports in YSR	140

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 sections for 'Asset Manager', 'Assistant Manager', 'CFDA', 'Client', 'Country', and 'District Manager'. Below these are buttons for 'Denominator', 'Department', 'Source', and 'Immediate Source'. On the far right, there are controls for 'DecimalDigits' (set to 0), 'Suppress Zero' (checked), 'Show Account' (set to 'Hide'), 'Detail' (checked), 'Summary' (unchecked), 'Tree Level' (set to 1), and 'Grid' (unchecked). A 'Freeze Pane' checkbox is also present. At the bottom, there is a preview grid with columns for 'Raleigh/Durham MTD', 'Raleigh/Durham YTD', 'Phoenix MTD', 'Phoenix YTD', 'Total MTD', and 'Total YTD'. The preview shows data for ASSETS, CURRENT ASSETS, CASH & CASH EQUIVALENT, and various totals.

	Raleigh/Durham MTD	Raleigh/Durham YTD	Phoenix MTD	Phoenix YTD	Total MTD	Total YTD
ASSETS						
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
LIABILITIES & EQUITY						

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.

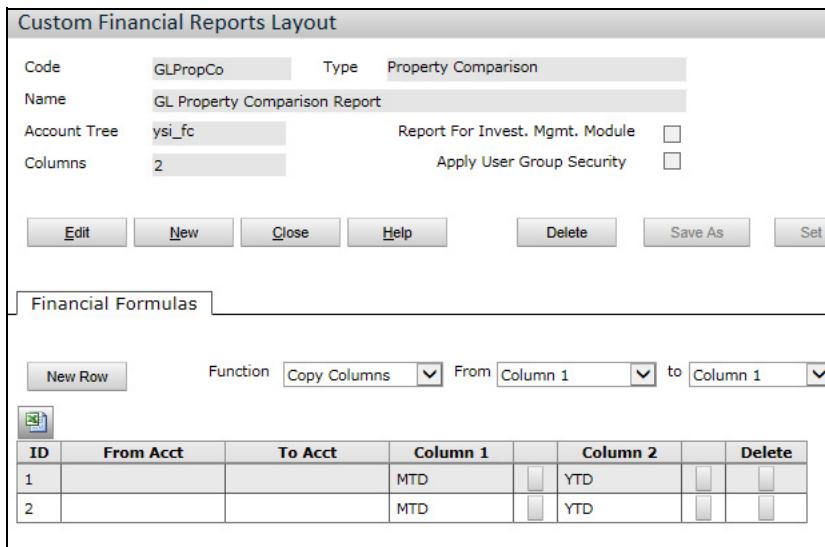
To set up a CFA Attribute Comparison report in YSR

1 Review your CFA report in Voyager.

The screenshot shows the 'Custom Financial Reports' dialog box. On the left, filters are applied: Property is set to 'comoff01^comoff', Book is 'accrual', Report Template is 'GLPropCo', Period is '02/2008 to 01/2017', Summary By is 'Asset Manager', and Account Tree is 'ysi_fc'. On the right, there are several dropdown menus for Denominator, Department, Source, Immediate Source, DecimalDigits (set to 0), Suppress, Show Account (set to Hide), and Summary (unchecked). Below these, there is a 'Freeze Pane' checkbox. The main area displays a table with columns for 'comoff01 MTD', 'comoff01 YTD', 'comoff02 MTD', 'comoff02 YTD', 'Total MTD', and 'Total YTD'. The table rows include 'Cash - Operating', 'Gross Potential Rent', 'Less: Vacancies', and 'Reserve Acct'. The preview shows numerical values for each category across the different time periods and property codes.

	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 x 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]
6				

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.

- 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).

Merged Report Setup

Code	cfa_glpropco	Notes																											
Description	CFA GL Property Comparison report for YSR																												
Select Statement																													
OR Script File																													
Key Column																													
Inactive	<input type="checkbox"/>	VoyagerPlus Report <input type="checkbox"/>																											
Save New		Define Filters Map Filters Dump SQL Dump Oracle Delete Setup																											
<input checked="" type="radio"/> Report Setup <input type="radio"/> Attachment & Email <input type="radio"/> Additional Roles <input type="radio"/> Output Options																													
<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 Filter</th> <th>Sections</th> <th>Inactive?</th> <th>Delete?</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>glpc</td> <td>GLPropComp.xlsx</td> <td>CFheader.txt</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 Filter	Sections	Inactive?	Delete?	1	glpc	GLPropComp.xlsx	CFheader.txt				<input type="checkbox"/>	<input type="checkbox"/>								<input type="checkbox"/>	<input type="checkbox"/>
Order	Report Code	Template File	Script File	Page Break Column	Map 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_glpropco)

Save Close

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

Merged Report: CFA GL Property Comparison report for YSR (cfa_glpropco)

Report Code: glpc

Save Close

Report Sections Attachment Setup

Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Relations	Key Columns	Primary?	Multiple Rows?
CF	CF		Custom Financials	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ColHead	Column Headers	PropHead		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>

7 Map filters for the Custom Financials data source.

Filter Mapping

Merged Report:	CFA GL Property Comparison report for YSR (cfa_glpropco)																				
Report Code:	glpc																				
SectionCode:	CF																				
Save Close																					
 Standard Report Filter <table border="1"> <thead> <tr> <th>Report Filters</th> <th>Constant Value</th> </tr> </thead> <tbody> <tr> <td>Template</td> <td>GLPropCo</td> </tr> <tr> <td>PropertyCode</td> <td>Property</td> </tr> <tr> <td>ToMMYY</td> <td>PostMonth</td> </tr> <tr> <td>FromMMYY</td> <td>PostMonth</td> </tr> <tr> <td>BookCodeList</td> <td>Book</td> </tr> <tr> <td>SuppressZero</td> <td>SuppressZero</td> </tr> <tr> <td>TreeCode</td> <td>ysi_fc</td> </tr> <tr> <td>Detail</td> <td>Detail</td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table>		Report Filters	Constant Value	Template	GLPropCo	PropertyCode	Property	ToMMYY	PostMonth	FromMMYY	PostMonth	BookCodeList	Book	SuppressZero	SuppressZero	TreeCode	ysi_fc	Detail	Detail		
Report Filters	Constant Value																				
Template	GLPropCo																				
PropertyCode	Property																				
ToMMYY	PostMonth																				
FromMMYY	PostMonth																				
BookCodeList	Book																				
SuppressZero	SuppressZero																				
TreeCode	ysi_fc																				
Detail	Detail																				



Map the **Book** filter field to the **BookCodeList** element in the **Standard Report Filter** 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 **Merged Report Setup** screen).

Filter Mapping

Merged Report:	CFA GL Property Comparison report for YSR (cfa_glpropco)				
Report Code:	glpc				
Save Close					
 Report Filters <table border="1"> <thead> <tr> <th>Report Filter Condition</th> </tr> </thead> <tbody> <tr> <td>Property (#Condition1#) Property.Hmy in (#Property#)</td> </tr> <tr> <td>Template (#Condition6#) #Template#</td> </tr> <tr> <td></td> </tr> </tbody> </table>		Report Filter Condition	Property (#Condition1#) Property.Hmy in (#Property#)	Template (#Condition6#) #Template#	
Report 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 headers. 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"/>																																			
Property Comparison Report <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 JOINAttributeName 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 hmy 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 hmys 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 hmvs, 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' interface in Voyager. The left pane contains various filters and settings: Property (moff01^comoff02), Book (accrual), Report Template (GLAttCo), Period (02/2008 to 01/2017), Comparison By (Market), and Account Tree (ysi_bs). The right pane shows a summary table with columns for Raleigh/Durham MTD, Raleigh/Durham YTD, Phoenix MTD, Phoenix YTD, Total MTD, and Total YTD. Below this is a detailed table for ASSETS, specifically CURRENT ASSETS, showing values for Cash - Operating, TOTAL CASH & CASH EQUIVALENT, TOTAL CURRENT ASSETS, and TOTAL ASSETS across the same six columns.

	Raleigh/Durham MTD	Raleigh/Durham YTD	Phoenix MTD	Phoenix YTD	Total MTD	Total YTD
ASSETS						
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' interface in Voyager. It displays the following configuration for a report named 'GL Attribute Comparison':

- Code: GLAttCo
- Type: Attribute Comparison
- Name: GL Attribute Comparison
- Account Tree: ysi_bs
- Columns: 2
- Report For Invest. Mgmt. Module:
- Apply User Group Security:

 Below this, there's a 'Financial Formulas' section with a table for defining formulas between columns. The table has columns for ID, From Acct, To Acct, Column 1, Column 2, and Delete. Two rows are present:

ID	From Acct	To Acct	Column 1	Column 2	Delete
1			MTD	YTD	<input type="checkbox"/>
2			MTD	YTD	<input type="checkbox"/>

- 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		

Sheet1 Sheet2

- 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 × 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="Total", "Total",)
5	&=[CF].[acctcode]	&=[CF].[acctDesc]	&=[CF].[column1]	&=[CF].[column2]
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).

Merged Report Setup

Code	cfa_glatto	Notes																											
Description	CFA GL Attribute Comparison report for YSR																												
Select Statement																													
OR Script File																													
Key Column																													
Inactive	<input type="checkbox"/>	VoyagerPlus Report <input type="checkbox"/>																											
<input type="button" value="Save"/> <input type="button" value="New"/> <input type="button" value="Define Filters"/> <input type="button" value="Map Filters"/> <input type="button" value="Dump SQL"/> <input type="button" value="Dump Oracle"/> <input type="button" value="Delete Setup"/>																													
<input type="button" value="Report Setup"/> <input type="button" value="Attachment & Email"/> <input type="button" value="Additional Roles"/> <input type="button" value="Output Options"/>																													
<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 Filter</th> <th>Sections</th> <th>Inactive?</th> <th>Delete?</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>glatto</td> <td>GLAttComp.xlsx</td> <td>CFHeader.txt</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 Filter	Sections	Inactive?	Delete?	1	glatto	GLAttComp.xlsx	CFHeader.txt				<input type="checkbox"/>	<input type="checkbox"/>								<input type="checkbox"/>	<input type="checkbox"/>
Order	Report Code	Template File	Script File	Page Break Column	Map Filter	Sections	Inactive?	Delete?																					
1	glatto	GLAttComp.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 Attribute Comparison report for YSR (cfa_glatto)																																																																																															
<input type="button" value="Save"/> <input type="button" value="Close"/>																																																																																																
<table border="1"> <thead> <tr> <th>Field Name</th> <th>Label</th> <th>Sequence</th> <th>Type</th> <th>Lookup Name</th> <th>Multi Select?</th> <th>List Values</th> <th>Code to ID</th> </tr> </thead> <tbody> <tr> <td>Property</td> <td>Property</td> <td>1</td> <td>Lookup List</td> <td>ysiPropertyOrListLookup</td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>CFType</td> <td>CF Type</td> <td>2</td> <td>List</td> <td></td> <td><input type="checkbox"/></td> <td>Attribute Comparison</td> <td></td> </tr> <tr> <td>PostMonth</td> <td>Post Month</td> <td>3</td> <td>Post Month</td> <td></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>Book</td> <td>Book</td> <td>4</td> <td>Lookup List</td> <td>ysiBookLookup</td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>SuppressZero</td> <td>SuppressZero</td> <td>5</td> <td>Checkbox</td> <td></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td>Template</td> <td>Template</td> <td>6</td> <td>List</td> <td></td> <td><input type="checkbox"/></td> <td>GLAttCo</td> <td></td> </tr> <tr> <td>AttributeSortName</td> <td>Summarize By</td> <td>7</td> <td>Lookup List</td> <td></td> <td><input type="checkbox"/></td> <td>SELECT A.sname codeField</td> <td>SELECT A.Hmy</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> </tbody> </table>									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"/>		
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 scope (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])`

- 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

Merged Report:	CFA GL Attribute Comparison report for YSR (cfa_glatco)																																								
Report Code:	glatco																																								
Report Sections																																									
<table border="1"> <thead> <tr> <th>Section Code</th> <th>Description</th> <th>SELECT Name</th> <th>Standard Report</th> <th>Map Filter</th> <th>Tokens</th> <th>Relations</th> <th>Key Columns</th> <th>Primary?</th> <th>Multiple Rows?</th> </tr> </thead> <tbody> <tr> <td>CF</td> <td>Custom Financials</td> <td></td> <td>Custom Financials</td> <td>▼</td> <td>■</td> <td>■</td> <td></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>ColHead</td> <td>Column Headers</td> <td>Attrhead</td> <td></td> <td>▼</td> <td>■</td> <td>■</td> <td></td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>▼</td> <td>■</td> <td>■</td> <td></td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> </tbody> </table>		Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Relations	Key Columns	Primary?	Multiple Rows?	CF	Custom Financials		Custom Financials	▼	■	■		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ColHead	Column Headers	Attrhead		▼	■	■		<input type="checkbox"/>	<input checked="" type="checkbox"/>					▼	■	■		<input type="checkbox"/>	<input checked="" type="checkbox"/>
Section Code	Description	SELECT Name	Standard Report	Map Filter	Tokens	Relations	Key Columns	Primary?	Multiple Rows?																																
CF	Custom Financials		Custom Financials	▼	■	■		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																
ColHead	Column Headers	Attrhead		▼	■	■		<input type="checkbox"/>	<input checked="" type="checkbox"/>																																
				▼	■	■		<input type="checkbox"/>	<input checked="" type="checkbox"/>																																

- Map filters for the Custom Financials data source.

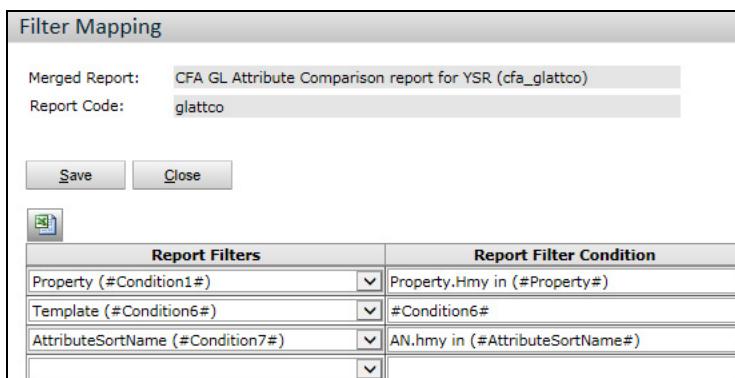
Filter Mapping

Merged Report:	CFA GL Attribute Comparison report for YSR (cfa_glatco)
Report Code:	glatco
SectionCode:	CF
Standard Report Filter	
Template	GLAttCo
PropertyCode	Property
FromMMYY	PostMonth
ToMMYY	PostMonth
BookCodeList	Book
SuppressZero	SuppressZero
AttributeSortName	AttributeSortName
AttributeSortValue	AttributeSortName
TreeCode	ysi_bs
Detail	1



Map the **Book** filter field to the **BookCodeList** element in the **Standard Report Filter** 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 **Merged Report Setup** screen).



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.
AttributeSortName	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.

Attribute Comparison Report		Raleigh/Durham MTD	Raleigh/Durham YTD	Phoenix MTD	Phoenix YTD	Total MTD	Total YTD
Acct Code	Acct Desc						
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	147
Smart Marker Parameters	149
Formula Smart Markers	151
Grouping and Subtotaling	153
Image Smart Markers	161
Details (Show, Hide, and Delete with YSROptions).....	163
Tips and Tricks	168

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 basic familiarity with smart markers and the steps for setting up an Excel template. For more information on these topics, see "Setting up Report Sections" on page 84.



This section explains how to write smart markers manually. If you install the Yardi Excel Add-In, you can also create smart markers with the YSR Smart Marker Designer. For more information about the YSR Smart Marker Designer, see Chapter 8, "Yardi Excel Add-In for YSR."

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 extra information (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.

Basic Smart Markers

Basic smart markers consist of &= followed by the data source (indicated by the section code) and the field name (column alias) of the Voyager data you want to display in your reports. For example, &=Party.FullName is a smart marker that represents the data in the FullName column of the data source (section code) referenced in the YSR report section coded Party.

If a user generates a report that retrieves multiple names from this section, Voyager expands the smart marker to create a list of full names in successive rows (assuming the section has **Multiple Rows** selected on the **Report Sections Setup**).

Basic smart marker formats include:

&=SectionCode.FieldName

&=[SectionCode].[Field Name]



Brackets are optional unless a name contains an embedded space. Then they are required.

Formula Smart Markers

You can also use smart markers to trigger Excel calculations like sums, averages, counts, and so on. Whereas Excel formulas refer directly to the contents of specific Excel cells, YSR formula smart markers refer indirectly to the data generated dynamically by the YSR report.

Formula smart marker formats include:

&=DynamicFormula

&=&=RepeatDynamicFormula



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.



For more information, see “Formula Smart Markers” on page 151.

Smart Marker Parameters

You can append *parameters* to any type of smart marker in order to control formatting options. Parameters appear in parentheses after the smart marker. For example, &=Party.FullName(skip:1) is a basic smart marker with a parameter that tells Voyager to skip one row for each row of data.



For more information about parameters, see “Smart Marker Parameters” on page 149.

Example Report Template and Report Output

This section shows an Excel template with different types of smart markers, followed by the resulting report output.

D7	=SUM(D5:D5)			
	A	B	C	D
1				
2				
3	Order Details			
4	Order ID	Quantity	UnitPrice	Total Amount
5	&=[Order Details].OrderID(group:normal)	&=[Order Details].Quantity	&=[Order Details].UnitPrice	&=&B(r)*C(r)
6				&=subtotal9:Order Details.OrderID
7				\$0.00
8				\$0!

Basic smart marker with grouping parameter

Dynamic formula smart marker

Repeating dynamic formula smart marker

Report output:

B8	=SUBTOTAL(9,B5:B7)		
	A	B	C
1			
2			
3	Order Details		
4	Order ID	Quantity	UnitPrice
5		12	\$14.00
6		10	\$9.80
7		5	\$34.80
8	10248	27	\$58.60
9			
10		9	\$18.60
11		40	\$42.40
12	10249	49	\$61.00

Smart Marker Parameters

In this section:

Supported Parameters	150
Using Parameters to Add Data in Alternating Rows.....	150

You can append *parameters* to any type of smart marker in order to control formatting options. Parameters appear in parentheses after the smart marker. For example, &=Party.FullName(skip:1) is a basic smart marker with a parameter that tells Voyager to skip one row for each row of data.

You can add 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.

Supported Parameters

YSR supports the following parameters:

noadd Prevents Voyager from adding extra rows to accommodate additional data.

skip:n Skips n number of rows for each row of data.

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.)

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 markets (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 parameters, 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.

Using Parameters to Add Data in Alternating Rows

You can combine the noadd and skip parameters to display different types of data in alternating rows. Because Voyager processes the template from bottom to top, add noadd on the first row to prevent Voyager from adding extra rows before the alternating row.

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	john.doe@sample.com		
3	Jane Doe		
4	jane.doe@sample.com		
5			

Formula Smart Markers

In this section:

Dynamic Formulas	152
Repeating Dynamic Formulas	152

Formulas enable you to insert Excel formulas into cells even when the formula refers to rows that expand during the report-generation process. Dynamic formulas generate calculations in a single cell; repeating dynamic formulas generate calculations in each of a set of rows.

Formula Options

You can add row information and parameters to formula tokens.

Example: &=&=A{r}/B{r}~(skip:1)

This example is a repeating dynamic formula that 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.



Use a tilde (~) to apply parameters to a formula.

Formulas support the following row offset options:

- 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

Dynamic Formulas

The following graphics illustrate a dynamic formula with offsets and the resulting report output.

	A	B	C	D	E
1	DATE	CTRL NUM	DESCRIPTION	AMOUNT	BALANCE
3	&=stmt.TranDate	&=stmt.TranCtrl	&=stmt.TranDec	&=stmt.TranAmt	&=&=SUM(D{r},E{-1})
4			CLOSING BALANCE		&=E3
5					

	A	B	C	D	E
1	DATE	CTRL NUM	DESCRIPTION	AMOUNT	BALANCE
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					

Repeating Dynamic Formulas

Use a repeating dynamic formula when you want to perform the same calculation on multiple lines. Repeating dynamic formulas begin with &=&=.

For example, the following graphics illustrate a repeating dynamic formula and the resulting report output.

	A	B	C
1	&=Answer.Price	&=Answer.Amount	&=&=A{r}*B{r}
2			

	A	B	C	D
1	100	2	200	
2	75.25	3	225.75	
3	25	5	125	
4				

Grouping and Subtotaling

In this section:

Grouping Parameters	153
Calculating Subtotals for a Single Data Group.....	154
Calculating Subtotals for Multiple Data Groups	155
Adding Labels to Subtotals.....	157

You can use grouping parameters to break your report data into sections. Grouping data by sections can make long reports easier to read and analyze, and it also allows you to perform calculations on each group of data. For example, you can provide sub-totals of different groups of account data, followed by a grand total at the end.

Grouping Parameters

There are three parameters for grouping report data:

group:normal The value represented by the grouped smart marker appears once per data group.

	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 Operates the same as **group:normal**, but merges the cells in the grouped section.

	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 Repeats the value represented by the grouped smart marker for each line of data in the group.

	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.

Calculating Subtotals for 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 report template in Excel. The main table has columns A, B, and C. Row 1 contains headers: 'Order ID' in A1, 'Quantity' in B1, and 'UnitPrice' in C1. 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 empty. Row 4 is also empty. Row 5 starts a detail section with a header row: 'Order ID' in A5, 'Quantity' in B5, and 'UnitPrice' in C5. It contains data for two groups: group 1 (rows 5-7) with OrderID 10248, Quantity 12, UnitPrice \$14.00, and a subtotal of \$58.60; and group 2 (rows 7-8) with OrderID 10249, Quantity 40, UnitPrice \$18.60, and a subtotal of \$61.00.

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	5	\$34.80
	5		► \$58.60
	6	9	\$18.60
	7	40	\$42.40
	8		► \$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.)

Calculating Subtotals for 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	A	B	C	D	E	F
Aging Detail Report						
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(group:)	&=Aging.propcode(group:)	&=Aging.leasecode(group:)	&=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	Aging Detail Report					
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						17,300.00
12						17,300.00
13						
14		nlp00002	t0001702	201200000007 201200000007	rent appfee	11,000.00 1,100.00
15						
16						12,100.00
17			t0001709	201200000028 201200000028 201200000028	rent capital fasb	5,500.00 7,700.00 1,100.00
18						
19						
20						14,300.00
21			t0001702	201200000010 201200000010	rent appfee	11,000.00 1,100.00
22						
23						12,100.00
24			t0001709	201200000032 201200000032 201200000032	rent capital fasb	5,500.00 7,700.00 1,100.00
25						
26						
27						14,300.00
28						52,800.00
29						
30						70,100.00
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 "Adding Labels to Subtotals" on page 157.

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 155.

Adding Labels to 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>&=Report.Year(Group:Repeat,skip:1)</code>	<code>&=Report.Date(co</code>	<code>&=Report.[Asset</code>		<code>&=Report.[Assessed</code>
			<code>Class](Group:Repeat)</code>	<code>&=Report.[Reported Cost]</code>	<code>Value]</code>
3				<code>&=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
Aging Detail Report					
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
Aging Detail Report						
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						
15						
16			Total for t0001702			12,100.00
17						
18						
19						
20			Total for t0001709			14,300.00
21						
22						
23			Total for t0001702			12,100.00
24						
25						
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.

Image Smart Markers

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. As 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 graphic:

```
//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		
4		&=charges.picturename(picture:scale100)

Image Smart Marker



As of Core Correspondence Plug-in 6, you cannot 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	ctaprop		
10	1000000049	300000049	10/2007	10/17/2007				Both	-1399999956	C-300000049	ctown	ctaprop		
13	1000000051	300000051	09/2007	10/17/2007				US Dollar	-299999944	C-300000051	ctown	ctaprop		
13	1000000051	300000051	09/2007	10/17/2007				US Dollar	-299999944	C-300000051	ctown	ctaprop		
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 input fields for 'Trans Type' (set to 'Journal'), 'Legal Entity' (set to 'alble'), 'Property' (empty), 'Account' (empty), 'From Date' (empty), 'To Date' (empty), and two checkboxes: 'Show Details' (unchecked) and 'Show Property Name' (unchecked). On the right, the 'Report Name' is set to 'Trans Register (TransReg)', 'Output Type' is 'Screen', and 'Merge Reports' is unchecked. There are also checkboxes for 'Attach Reports', 'Email Reports', and 'Show on Portal'. Below these are 'Generate' and 'Clear' buttons. At the bottom, there are tabs for 'Register Reports', 'Show Detail' (which is selected and highlighted in blue), and 'YSROptions'. The main area displays a table titled 'Transaction Register' with the following data:

Doc. Seq. No.	Charge To	Batch	Control	Legal Entity	Property	Charge Code	Base Curr.	Net Amount	Vat Amount	Gross Amount	Amount Paid	Refer
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		

Below the table, a note says 'Excel columns B-H are hidden'.

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"/>
TToDate	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(&=TransR.C&=TransR.T&=TransR.F&=TransR.T&=TransR.BookName)						
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 165.

- 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								&=YSRFilterValues.ShowDetail

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.

Tips and Tricks

In this section:

Adding Special Formatting to Expanding Row Data	168
Date and Number Formatting	169
Freeze Panes.....	171
Drill-Down Links	172

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"/>
Report Columns	Actual	Summary <input type="checkbox"/>	Tree Level 1	Show Acct. Code <input checked="" type="checkbox"/>	Source	
		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	Rept - 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	
	Total	0.00	801,140,395.78	801,139,910.46	485.32	

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.

	A	B	C	D	E	F	G	H	I	J
1		Actual	Budget	Variance						
2	&=Region.acctdesc	&=[Region].[column1]	&=[Region.column2]	&=[Region].[column3]						
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										



This example is only one way to use conditional formatting in Excel. For more information about conditional formatting, see Microsoft Help.



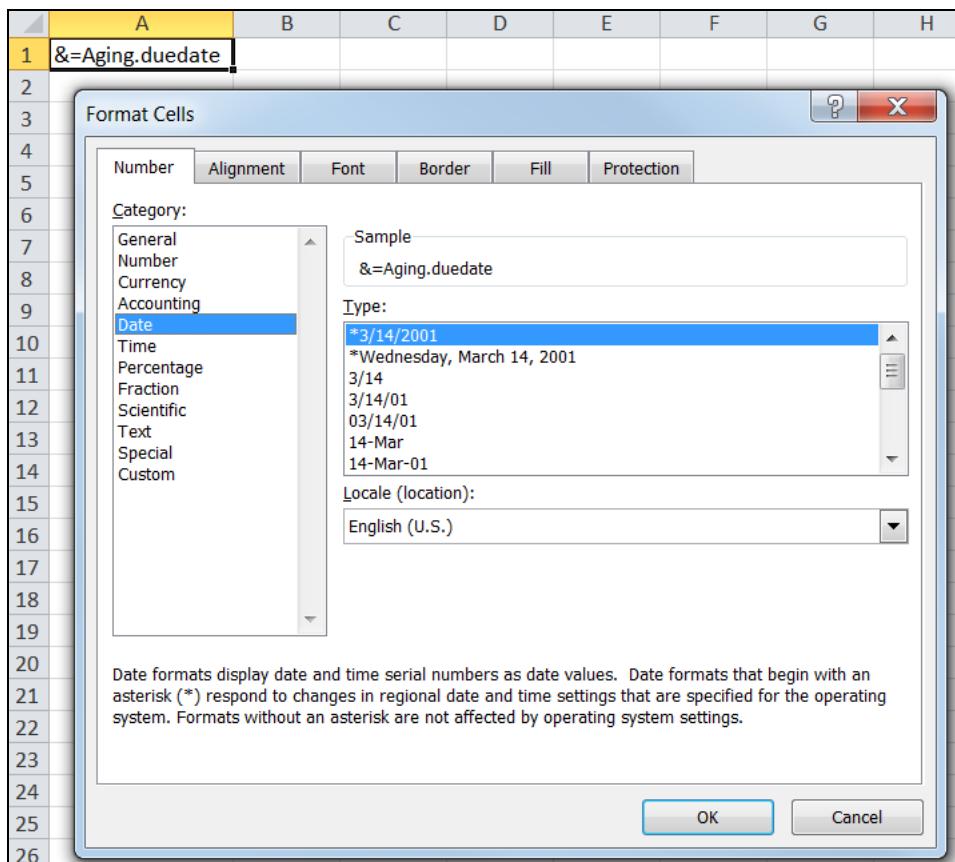
The preceding example uses conditional formatting in Excel 2010. The options available to you depend on the version of Excel you are using.

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' list, 'Number' is chosen. The 'Decimal places' dropdown is set to 2. Under 'Negative numbers', the sample shows '-1234.10' in blue, '1234.10' in red, '(1234.10)' in blue, and '(1234.10)' in red. A note at the bottom states: 'Number is used for general display of numbers. Currency and Accounting offer specialized formatting for monetary value.' The 'OK' button is highlighted.

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' appears, stating: 'Keep rows and columns visible while the rest of the worksheet scrolls (based on current selection)'. Below this, two sub-options are listed: 'Freeze Top Row' and 'Freeze First Column', each with a brief description of their function.

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:
`¶meter1=X¶meter2=Y¶meter3=Z`

For example, the following graphic shows a column containing a link to the tenant screen:

The screenshot shows an Excel spreadsheet with a table. The table has columns labeled A through G. Column D contains the formula `&=&=HYPERLINK("javascript:DrillDown('../pages/TenantSwitch.aspx?1=1&TenantID='&E{r}&'')",F{r})`. The formula bar at the top of the spreadsheet also displays this formula. The table data includes columns for Property, PropID, PropName, Tenant, LeaseID, LeaseName, and Units.



If you want to retrieve the URL of your webshare for use in hyperlinks, use this token: `#@@WEBSHARENAME#`. For more information about using tokens in scripts, see "Session and User-Related Tokens" on page 38.

CHAPTER 8

Yardi Excel Add-In for YSR

In this chapter:

Yardi Excel Add-In Overview.....	174
Building YSR Report Templates with the Add-In	176
Building YSR Reports with the Add-In (Dump SQL)	189

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 both a Smart Marker Designer, which you can configure to identify and read from both custom SQL scripts and Voyager analytics, and a report-building function (Dump SQL) that creates a package containing all YSR setup elements.



The Yardi Excel Add-In supports both YSL.NET (Yardi Spreadsheet Link) and YSR. This document describes the menu items pertinent to YSR. For information about the menu items that support the design activities of YSL.NET, see the *Yardi Spreadsheet Link User's Guide*.



You can find the Yardi Excel Add-In on Client Central (search for **Excel Add**). For information about installing the Add-In, see "Yardi Spreadsheet Link (YSL.NET) Installation and Setup" in the *Yardi Spreadsheet Link User's Guide*.

Yardi Excel Add-In Overview

With the Yardi Excel Add-In, you can design a YSR report without using the Voyager interface. The Add-In also contains the YSR Smart Marker Designer, which helps you write smart markers and format smart marker parameters.

The Add-In is available on Client Central under Enterprise and Voyager Software Resources. Most of the Add-In features support Yardi Spreadsheet Link, but the Add-In also contains tools for YSR. You can find documentation for the YSL-related features of the Add-In in the *Yardi Spreadsheet Link User's Guide*.

Advantages of the Yardi Excel Add-In

With the Yardi Excel Add-In, you can:

- Create a YSR report without using the Voyager YSR interface.
- Dump SQL: create a .pkg file containing all the design specifications for your YSR report.
- Format smart markers automatically with the YSR Smart Marker Designer.
- Retrieve the field aliases used in Voyager analytics (necessary for building smart markers for any report that retrieves data with Voyager analytics).
- Check your Excel template for errors.

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** 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.....	176
Enabling the YSR Smart Marker Designer	176
Selecting Data Sources.....	178

This section describes how to use the Yardi Excel Add-In to prepare report templates for YSR reports.

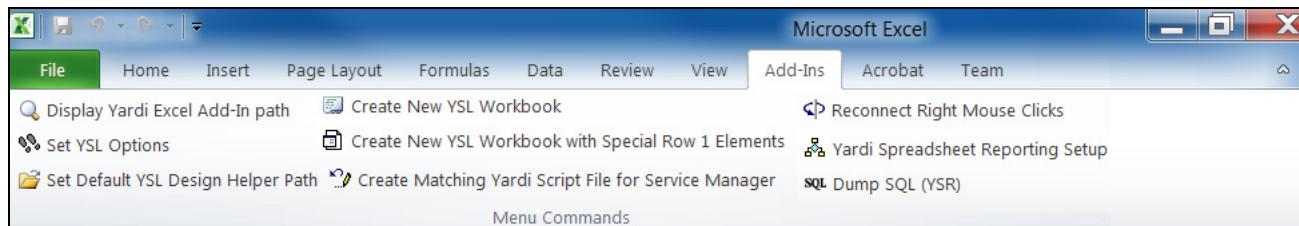
Downloading and Installing the Yardi Excel Add-In

To use the Yardi Excel Add-In, first download the Add-In from Client Central and install it. To find the Add-In, search Client Central for **Excel Add**.



Refer to Microsoft Excel Help for more information about how to install add-ins or see "YSL.NET Installation Procedures" in the *Yardi Spreadsheet Link User's Guide*.

After you download and install the YSR Excel Add-In, new options appear on the **Add-Ins** tab on the Excel ribbon.



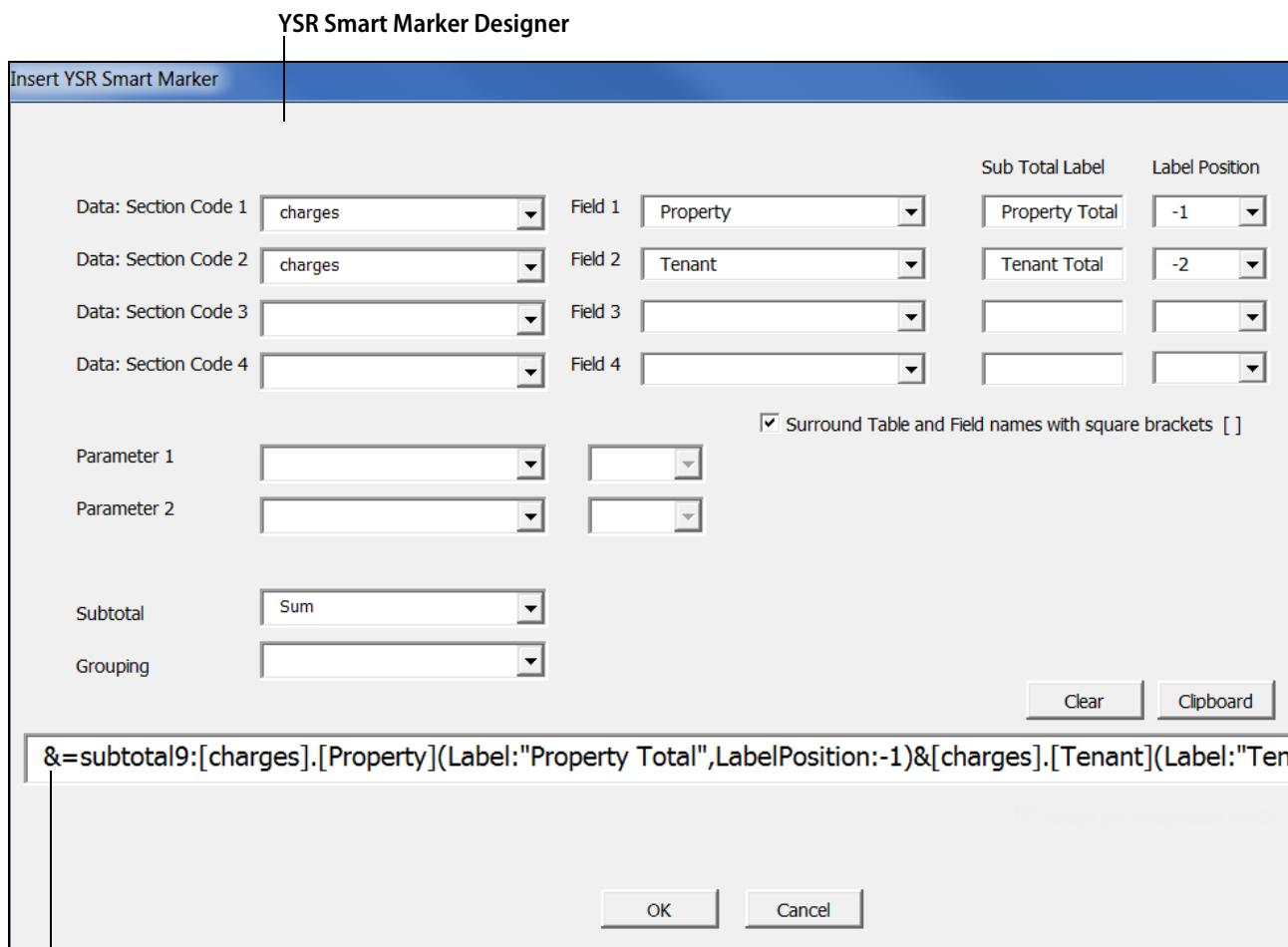
This graphic illustrates the Add-In menu set in Excel 2010. Your screen may appear differently depending on your version of Excel and the other Add-Ins that you have installed (if any).

The last two items, **Yardi Spreadsheet Reporting Setup** and **Dump SQL (YSR)** contain all the tools for creating a YSR report.

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.

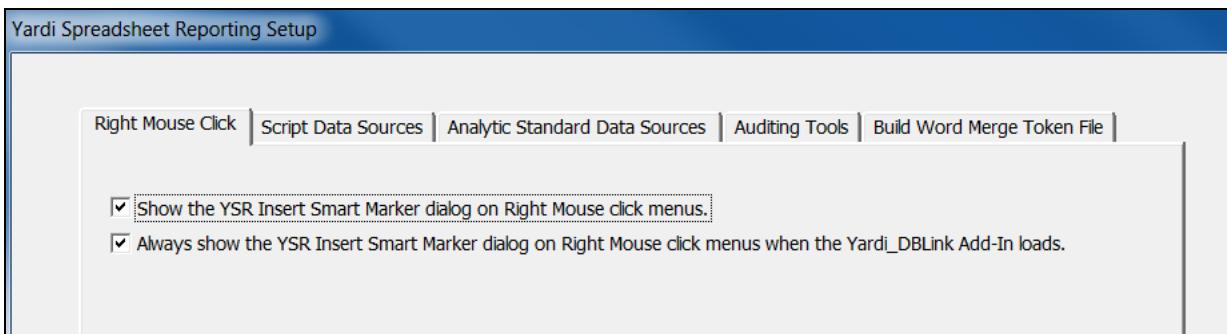


Smart marker in process of being built

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.

2 Select the Right Mouse Click tab.



- 3** 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 181.

Yardi Excel Add-In Settings and Configuration

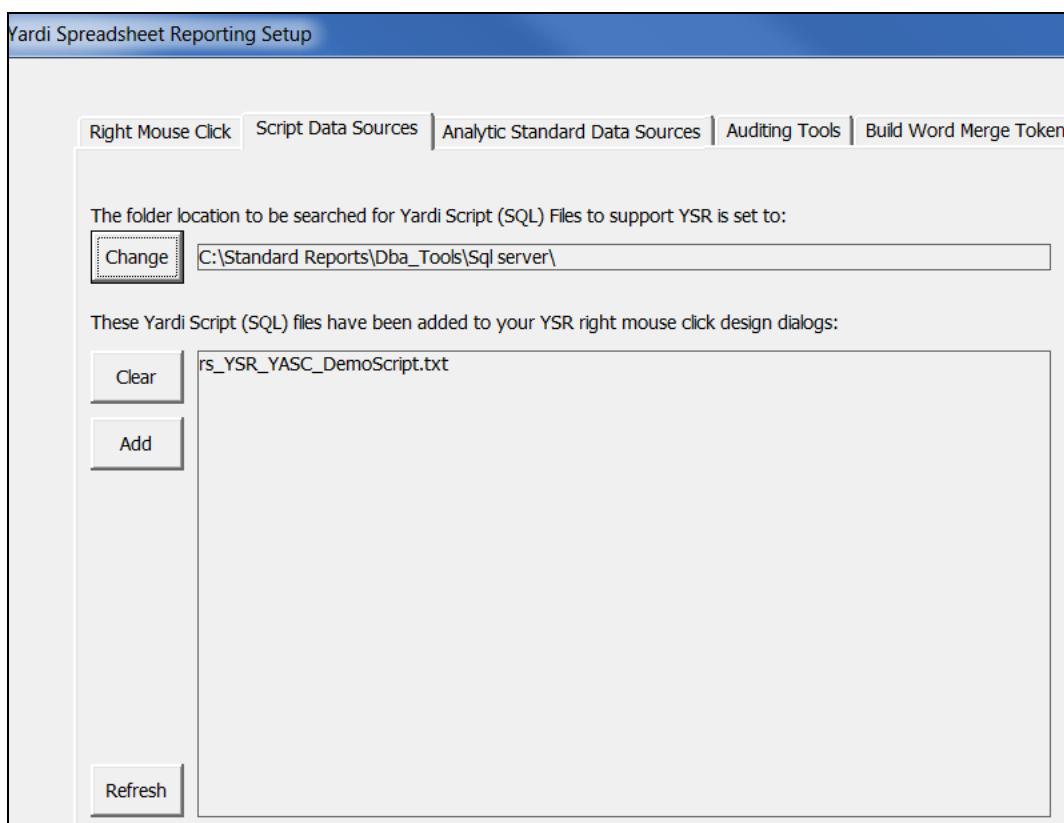
All settings and configuration data related to the Yardi Excel Add-In are specific to your computer. If you change computers, you must reconfigure the Yardi Excel Add-In.

This section includes the following procedures:

To select a custom SQL script as a data source.....	179
To refresh a script data source	181
To select a Voyager analytics report as a data source.....	182

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.

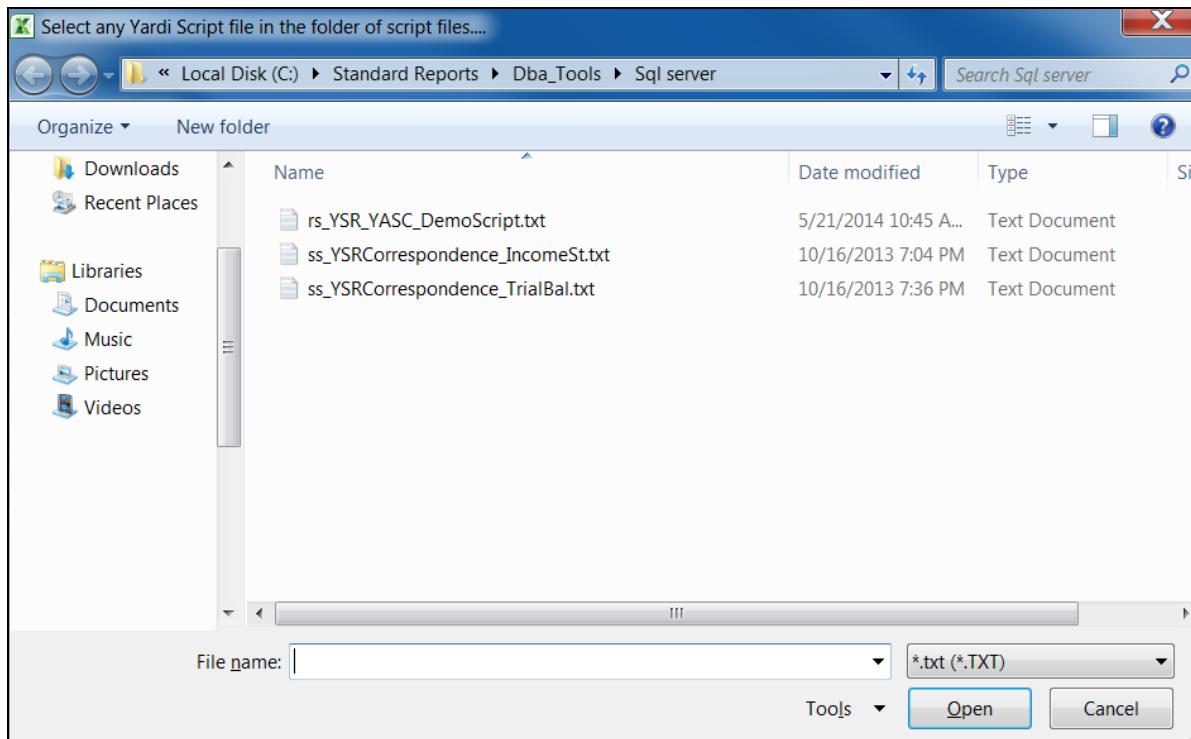
- 5 Select a file and click **Open**. The filename of the script appears in the main box of the **Script Data Sources** tab, and the file path of the file appears in the **Change** field.



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.



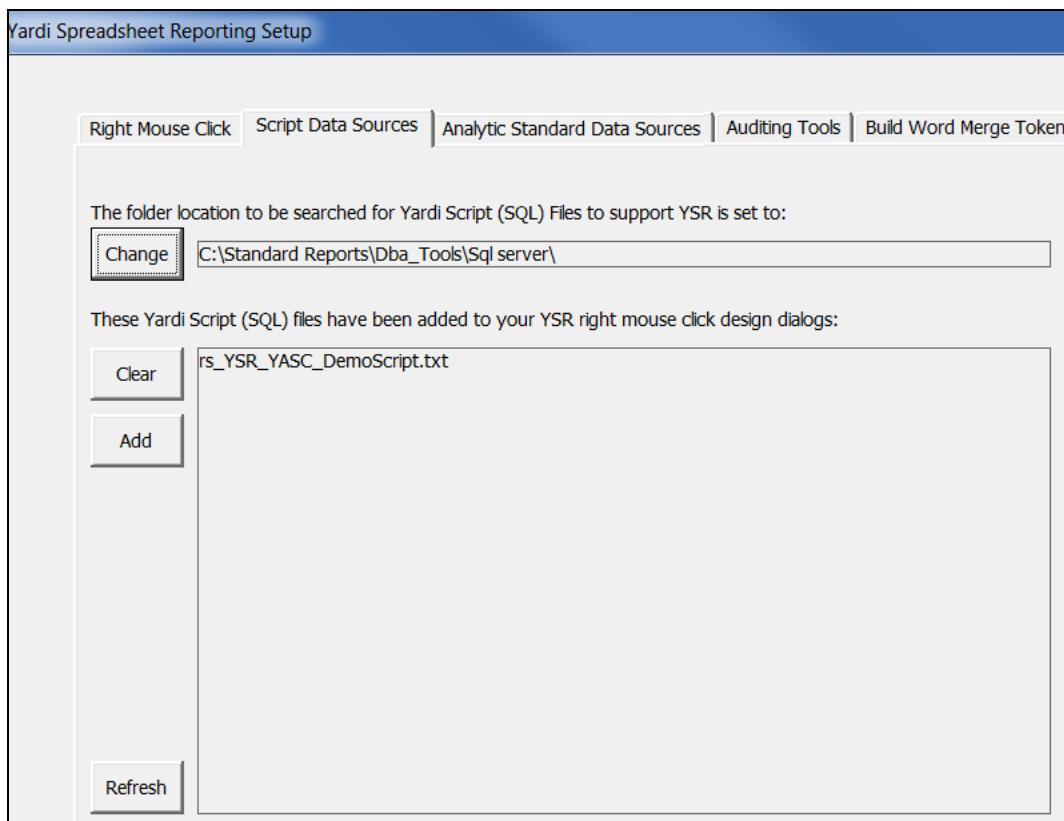
- b Navigate to the script files that accompany your report.
- c Select a file and click **Open**. The file appears in the main section of the **Script Data Sources** tab of the **Yardi Spreadsheet Reporting Setup** screen.
- d Repeat this process to add additional script files.
- 7 Click **Ok**. The 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 number of Voyager analytics reports available in YSR is increasing. Download and install a new version of the Yardi Excel Add-In regularly to take advantage of the most recently available analytics reports.



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.

- From the Excel top menu, select **Add-Ins > Yardi Spreadsheet Reporting Setup**. The **Yardi Spreadsheet Reporting Setup** screen appears.

- Select the **Analytic Standard Data Sources** tab.

Yardi Spreadsheet Reporting Setup

Right Mouse Click | Script Data Sources | **Analytic Standard Data Sources** | Auditing Tools | Build Word Merge Token File |

The checked Analytic Data Source will be added to your YSR right mouse click design dialogs, and aliased for 'Section Code' using any text you include in the 'Section Code' field.

<input type="checkbox"/> AP Analytics * (ss_YSRInterface_AP_2475552.pkg)	<input type="checkbox"/> CRM Analytics * (ss_Unknown.PKG)
<input type="checkbox"/> AR Analytics * (ss_YSRInterface_Financial.pkg)	<input type="checkbox"/> Custom Financials (ss_YSRInterface_CustomFinancials.pkg)
<input type="checkbox"/> AR Analytics Domestic * (ss_YSRInterface_Financial.pkg)	<input type="checkbox"/> Custom IM Data (ss_Insert_IM_YSR_Object_CusData.pkg)
<input type="checkbox"/> Commercial Analytics * (ss_YSRInterface_Commercial.pkg)	<input type="checkbox"/> Financial Analytics (ss_YSRInterface_Financial.pkg)
<input type="checkbox"/> Condo Accounts Receivable Aging * (ss_CondoReportsYSRInterface.pkg)	<input checked="" type="checkbox"/> Generic Performance Analytics (ss_Unknown.PKG)
<input type="checkbox"/> Condo Correspondence Invoice * (ss_CondoReportsYSRInterface.pkg)	<input type="checkbox"/> IM Attribute Data (ss_Insert_IM_YSR_Object_CusData pkg)
<input type="checkbox"/> Condo Correspondence Ledger * (ss_CondoReportsYSRInterface.pkg)	<input type="checkbox"/> IM Capital Roll Forward Analytics
<input type="checkbox"/> Condo Correspondence Letter * (ss_CondoReportsYSRInterface.pkg)	<input type="checkbox"/> IM Custom Financial Data
<input type="checkbox"/> Condo Correspondence Purchase * (ss_CondoReportsYSRInterface.pkg)	<input type="checkbox"/> IM Custom1To1 Data (ss_Insert_IM_YSR_Object_CusData pkg)
<input checked="" type="checkbox"/> Condo Owner Ledger Analytic * (ss_CondoReportsYSRInterface.pkg)	<input type="checkbox"/> IM Data (ss_Insert_IM_YSR_Object_CustomCorrespondenceData.pkg)
<input type="checkbox"/> CORRESP_Const_DrawAnalytics * (ss_iCMW_YSRCorrespondence.pkg)	<input type="checkbox"/> IM GL Data (ss_Insert_IM_YSR_Object_CustomCorrespondenceData pkg)
<input type="checkbox"/> CORRESP_Const_JobCostAnalytics * (ss_iCMW_YSRCorrespondence.pkg)	<input type="checkbox"/> IM Performance Data (ss_Insert_IM_YSR_Object_PerformanceData.pkg)
<input type="checkbox"/> CORRESP_Maint_FixedAsset * (ss_Maint_YSRCorrespondence.pkg)	<input type="checkbox"/> International Residential Analytics * (ss_IntRes_YSRInterface_FinancialData.pkg)
<input type="checkbox"/> CORRESP_Maint_FixedAsset_DataTable (ss_Maint_YSRCorrespondence.pkg)	<input type="checkbox"/> Intl Transaction Registers * (ss_YSRInterface_FinancialData.pkg)
<input type="checkbox"/> CORRESP_Maint_Inventory * (ss_Maint_YSRCorrespondence.pkg)	<input type="checkbox"/> PR Preferred Returns Analytics (ss_Unknown.PKG)
<input type="checkbox"/> CORRESP_Maint_Inventory_DataTable (ss_Maint_YSRCorrespondence.pkg)	<input type="checkbox"/> Residential Analytics * (ss_YSRInterface_ResidentialAnalytics.pkg)
<input type="checkbox"/> CORRESP_Maint_PlannedMaintenance * (ss_Maint_YSRCorrespondence.pkg)	<input type="checkbox"/> Senior Community Analytics (ss_YSRCorrespondenceCommunityData.pkg)
<input type="checkbox"/> CORRESP_Maint_WorkOrder * (ss_Maint_YSRCorrespondence.pkg)	<input type="checkbox"/> Senior Listing Analytics (ss_YSRCorrespondenceListingData.pkg)
<input type="checkbox"/> CORRESP_Maint_WorkOrder_DataTable (ss_Maint_YSRCorrespondence.pkg)	<input type="checkbox"/> Tenancy Schedule * (ss_YSRInterface_CommercialTenancyScheduleData.pkg)
<input type="checkbox"/> Correspondence (International) * (ss_YSR Correspondence Interface.pkg)	

* Analytic can be extended by authoring custom tokens.

- Select the check box corresponding to the Voyager analytics report used in your YSR report.
- 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 “Formula Smart Markers” on page 151 and “Image Smart Markers” on page 161.

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.

Section Code	Description	SELECT Name	Standard Report	Map Filter
finsum	Fin Summary	finsum	<input type="button" value="▼"/>	<input type="button" value=""/>
			<input type="button" value="▼"/>	<input type="button" value=""/>

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).

To create a smart marker with the YSR Smart Marker Designer

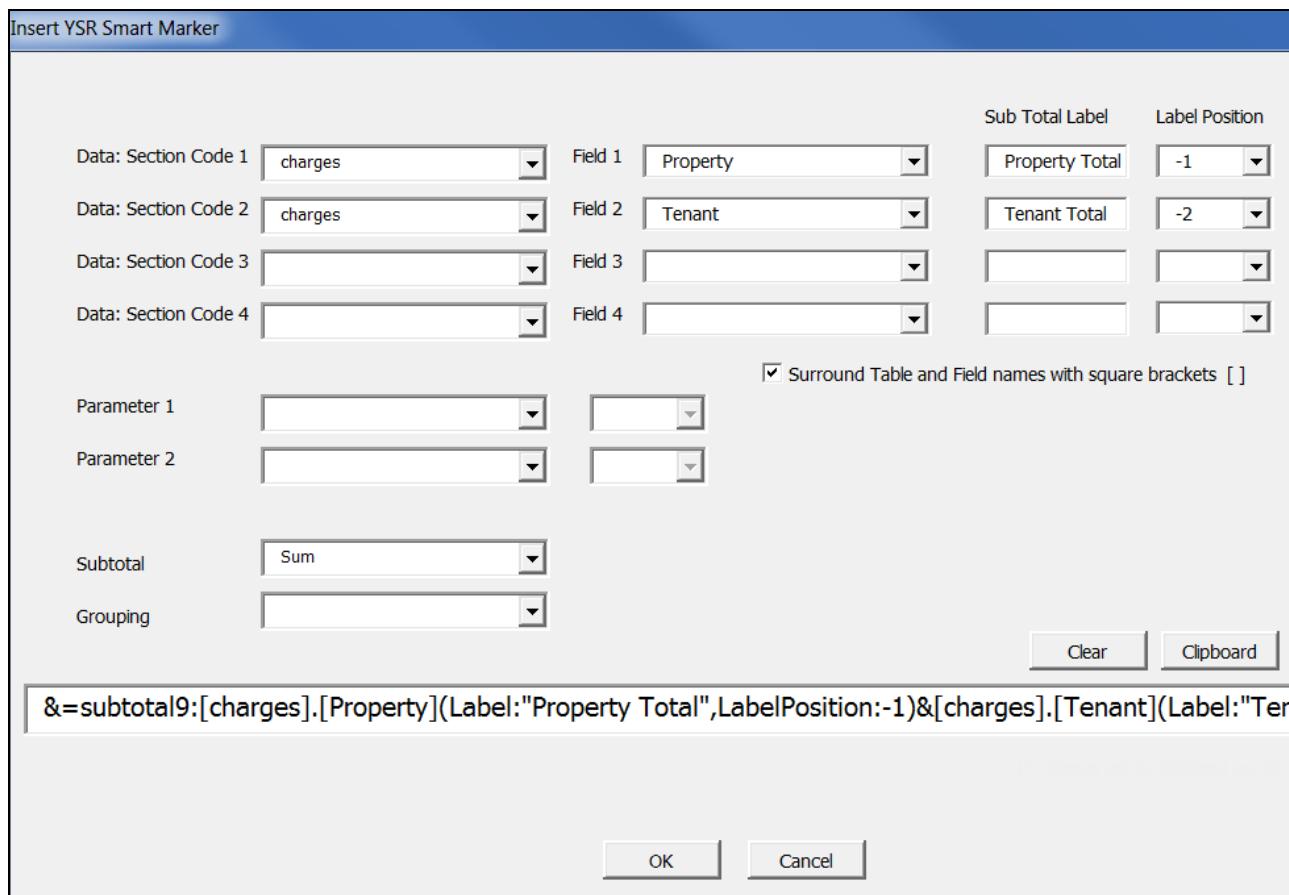
- 1 In your Excel template, right-click in any worksheet cell. A set of right-click menu options appears.
- 2 Select **YSR Smart Marker**. The **Insert YSR Smart Marker** screen appears.
- 3 Complete the fields. For more information, see “Insert YSR Smart Marker Screen Reference” on page 184.



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 178.

For more information about the difference between section names and codes, see “Select section names and section codes” on page 183.

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 “Calculating Subtotals for Multiple Data Groups” on page 155.

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 “Calculating Subtotals for Multiple Data Groups” on page 155.
Subtotal Label	For use with subtotal smart markers. The label of the subtotal.
Label Position	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.
Surround Table and Field Names with square brackets []	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. TIP You must select this option if your section codes and field names contain embedded spaces. TIP 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.
Parameter 1-2	Formatting options. For more information about each option, see “Smart Marker Parameters” on page 149. TIP 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 Parameter 1 field to make additional specifications. For example, select 2 to skip two rows.
Subtotal	The Excel function performed by the smart marker (optional). TIP 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 “Calculating Subtotals for a Single Data Group” on page 154. NOTE 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 “Calculating Subtotals for Multiple Data Groups” on page 155.
Grouping	The method YSR uses to cluster repeated data items. For examples and more information, see “Grouping Parameters” on page 153.
Clipboard	Copies the smart marker to your clipboard so that you can paste it elsewhere.

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.

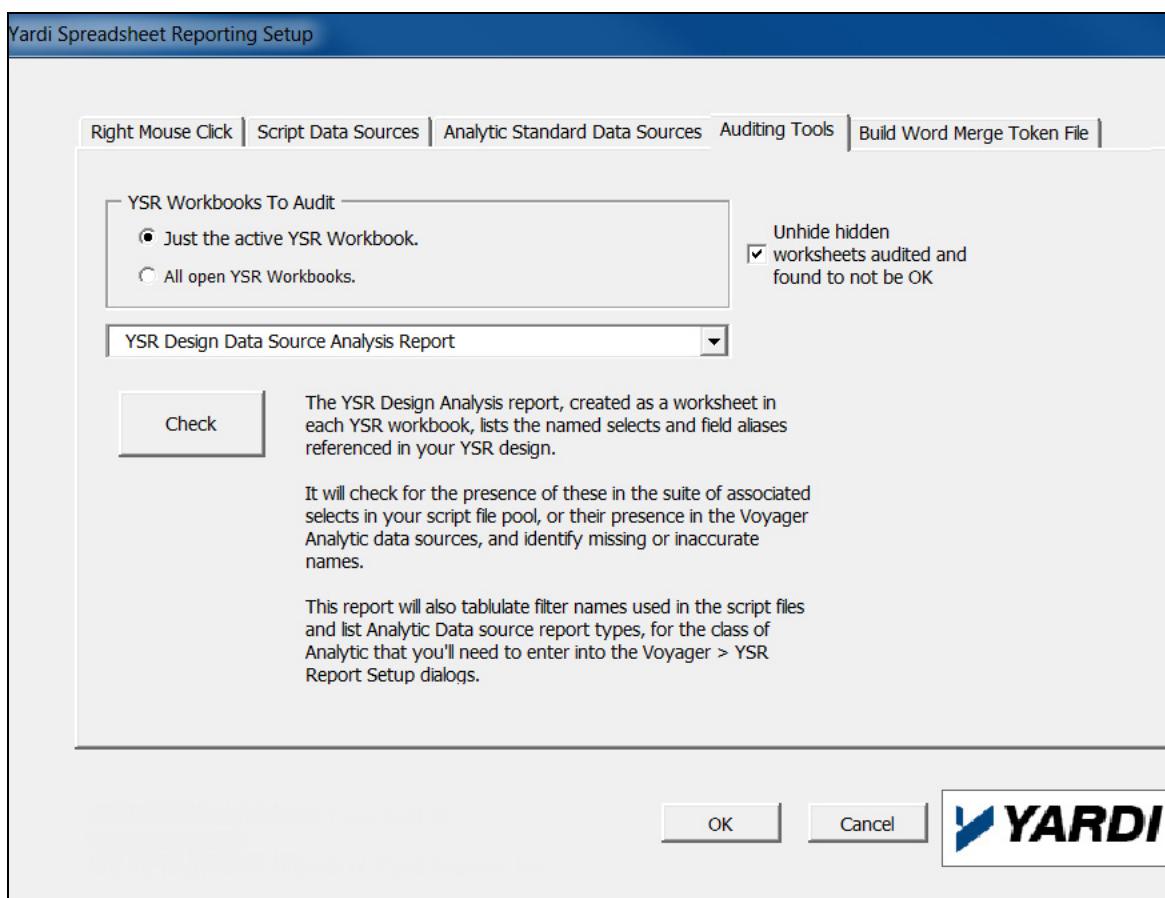
The Yardi Excel Add-In displays the audit results in your Excel template, in a new worksheet titled **YSR Design Analysis**. For example, the following graphic shows the results of an audit in which there are five correctly formatted smart markers and one problematic smart marker. You can link directly 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 Sel
1	OK Analytic or Named Select.			
2	Field Name not found.	ASCAP17	\$E\$2	ASCAP17
3	OK	ASCAP12	\$A\$2	ASCAP12
4	OK	ASCAP12	\$AA\$2	ASCAP12
5	OK	ASCAP12	\$AB\$2	ASCAP12
6	OK	ASCAP12	\$AC\$2	ASCAP12
7	OK	ASCAP12	\$AD\$2	ASCAP12

To check an Excel template for errors

- 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.



3 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 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.

Building Merge Token Files for Word Templates

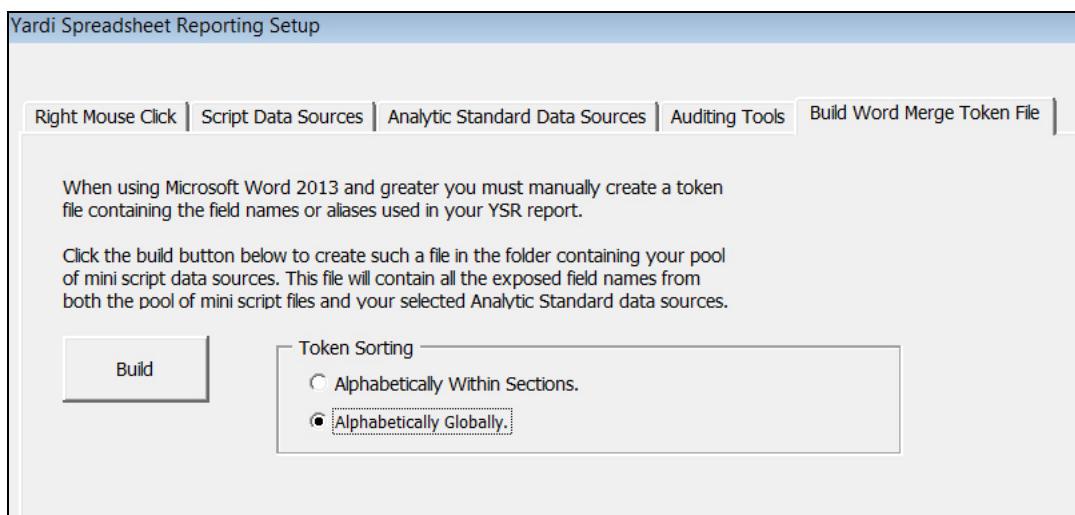
To prepare Word report templates, you must use Word merge fields to stand in for Voyager data. With some versions of Word, you can insert merge fields manually. With Word 2013, however, you must link to a data source before you can insert merge fields.

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 82.

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.

Alphabetically Within Sections	When you view the token list in Word, tokens are sorted by section. Within each section, tokens are listed alphabetically.
Alphabetically Globally	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 82.

Building YSR Reports with the Add-In (Dump SQL)

In this section:

Preparing Script Files For Use With the Yardi Excel Add-In.....	190
Adding a Top-Level Select Statement.....	190
Building the //Filter Section	191
Filter Mapping and the Value1 Parameter.....	194
Val1 Parameter Conventions	194
Building Report Packages (Dump SQL)	195

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 graphic shows a basic YSR script with one select statement named *Main*:

```
//Select Main
SELECT K.dOccupiedArea OccupiedAreaSum,
K.dtDate as 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 37.

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 194 and “Val1 Parameter Conventions” on page 194.



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 191.

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.



If you use a top-level select statement, it must retrieve a data field that can function as a primary key. Then, you must identify the primary key by making a selection from the **Key Column** field on the **Dump SQL** tab of the **Yardi Spreadsheet Reporting Dump SQL** screen.



For background information about the top-level select statement, see “Top-Level Select Statements” 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:

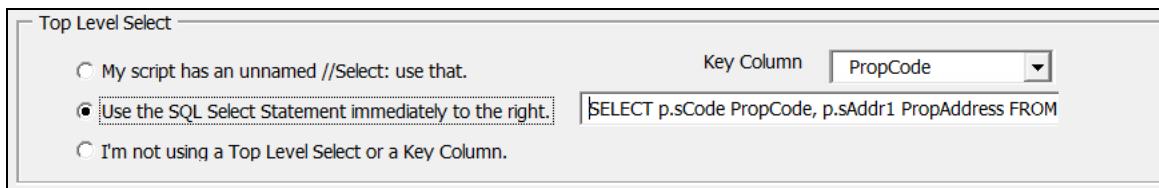
- Add an unnamed select statement to your script file. For example:

```
//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 191 and “Filter Mapping and the Value1 Parameter” on page 194.

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:

The screenshot shows the 'Custom Financial Income Statement' dialog box. On the left, under 'Custom Filter Fields', there is a table with the following data:

Property	yfund
Account Tree	
Books	accrual
From Date	01/2000
To Date	12/2014
Suppress Zero	<input type="checkbox"/>
Is Consolidate	<input type="checkbox"/>

On the right, the 'Report Name' is set to 'Custom Financial Income Statement (CF_IS)'. Other settings include 'Output Type' (PDF), 'Merge Reports' (unchecked), 'Show Grid' (unchecked), and 'Generate' and 'Clear' buttons.

- 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,
    p.hMy I,          Property,      Property,      ,3,
    ,Y,              ,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:

The screenshot shows the 'Build your Script Filter' interface. At the top, there are tabs: 'Dump SQL', 'Build your Script Filter' (which is selected), and 'Build your Analytics Filter'. Below the tabs are two buttons: 'Commit Changes to Script' and 'Rebuild Grid from Script File'.

The main area displays a grid of filter elements:

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"/>

At the bottom of the grid, there is a 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, you must build them in Voyager. For more information, see "YSR Report Filter Definition" on page 51.

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 Custom SQL Sub-Reports" on page 69.

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 194.

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 197.

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.

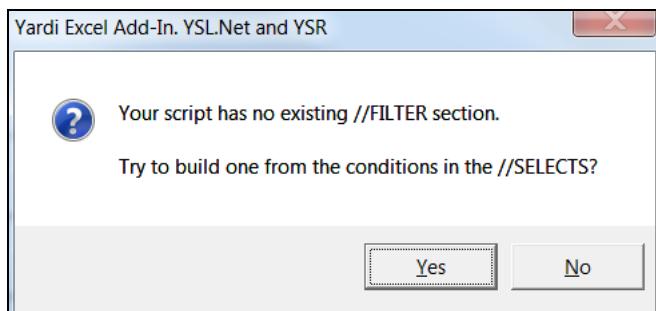
To build a YSR report with the Yardi Excel Add-In

- 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 179.



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 197.
 - 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 198.
 - 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.

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)

Merge Report Code: CommOc1 YSR Excel Template :Copy of YSR_OccupiedAreaOverTime.xlsx

Description: Commercial Occupied Area

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.

Create matching script file for Report Scheduler.

YARDI

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.

Top-Level Select section	Options for handling the top-level select statement.
	<p>My script has an unnamed //Select: use that 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 Key Column field to the right.</p> <p>Use the SQL Select Statement immediately to the right Enter a select statement in the field to the right. You must also specify the primary key in the Key Column field.</p> <p>I'm not using a top-level Select or a Key Column 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 “Top-Level Select Statements” on page 23.</p>
Key Column	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>NOTE 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 Key Columns” on page 25.</p>
Create matching script file for Report Scheduler	<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 “Making YSR Reports Available to Report Scheduler” on page 104.</p>

Build Your Script Filter Tab Screen Reference

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^Elimir	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)

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.

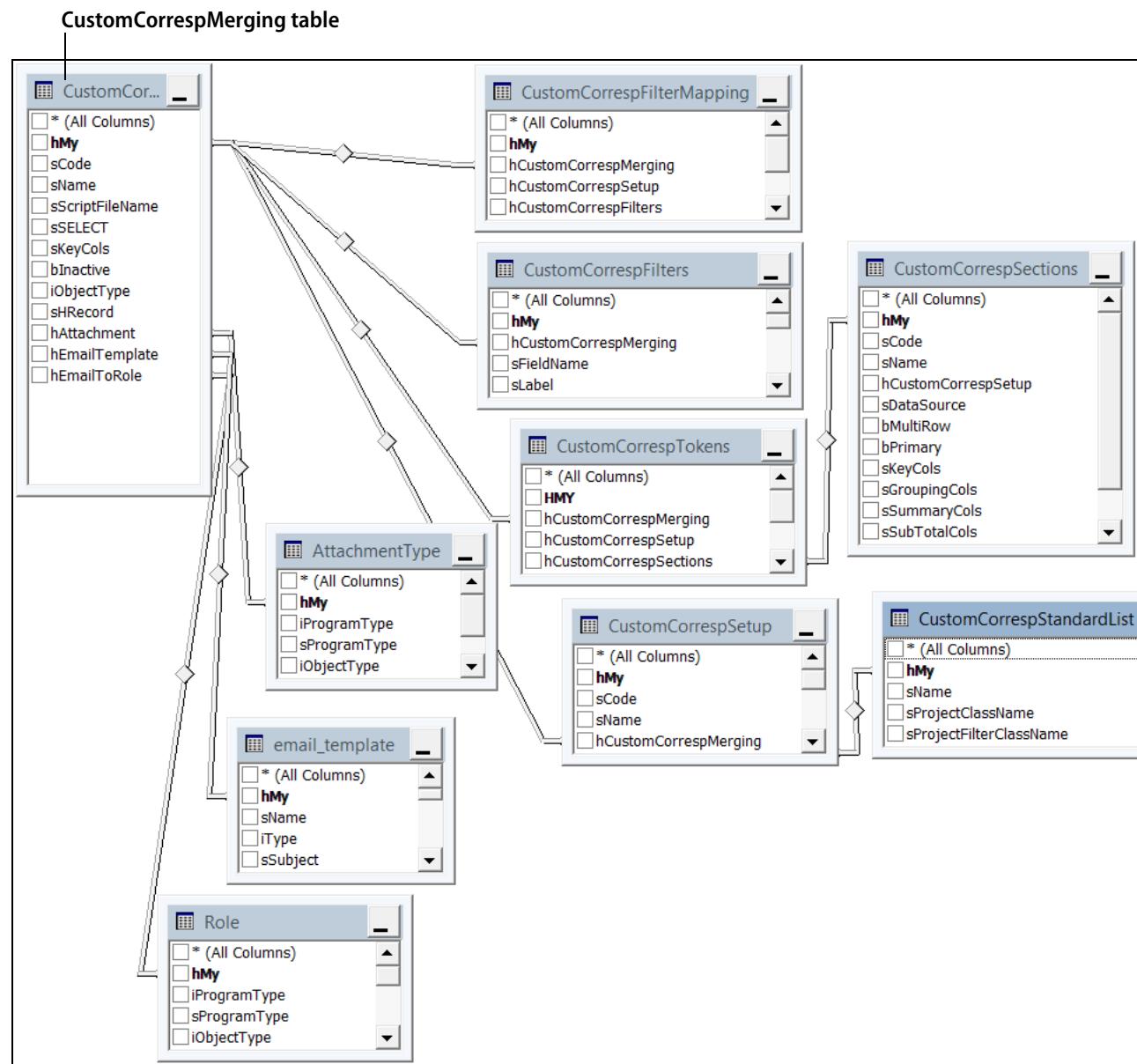
Element Type	The type of filter element. For more information about filter elements, see Chapter 5, "Vista Filters" in the <i>Yardi SQL Scripting Guide</i> .
Data Type	The type of data the filter accepts. For more information about data types, see Chapter 5, "Vista Filters" in the <i>Yardi SQL Scripting Guide</i> .
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	<p>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.</p> <p>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.</p> <p>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.</p>
Val1	<p>A small section of SQL appended to the executing script data source to introduce the filtration specified by the user.</p> <p>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: p.hmy in (#Property#).</p> <p>TIP 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.</p> <p>NOTE 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 194.</p> <p>For more information about Value1 clauses, see Chapter 5, "Vista Filters" in the <i>Yardi SQL Scripting Guide</i>.</p>
Mandatory	Makes the filter field a required field.
Multi-Select	Users can filter for multiple items.
Parent	Indicates that this filter field is restricted by values in another filter field. Enter the sequence number of the parent filter field.

Automatically Rename Named Tokens to Numbered Conditions in Script File on Dump SQL (recommended on first use)	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 Dump SQL : WHERE 1=1 #Property# #Tenant# becomes WHERE 1=1 #Condition1# #Condition2# TIP 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 data structure, or schema, of a YSR report.



APPENDIX A

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

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' when 'Print Forms-Copy A' 1098_YSR
then 'Form_CopyA' when 'Print Forms-Copy B' then 'Form_CopyB' when 'Print Forms-Copy C' then
'Form_CopyC' when 'Form 1096' then 'Form_1096' end)+'_Template.xlsx'
```

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 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 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 Tenant Data Collection - City Of Chicago (AFFTDC)

Top-Level Select Script Files

YSR Script Files

rs_Aff_TIC_TenantDataCollectionCityOfChicago.txt

AFFTDC

YSR Template Names

rx_Aff_TIC_TenantDataCollectionCityOfChicago.xls

AFFTDC

Audit NAHMA Fields (AffNAHMAAudit)

Top-Level Select Script Files

YSR Script Files

rs_Aff_AuditNAHMAFields.txt

EventDet

rs_Aff_AuditNAHMAFields.txt

HHMem

rs_Aff_AuditNAHMAFields.txt

Income

rs_Aff_AuditNAHMAFields.txt

Assest

YSR Template Names

rx_Aff_AuditNAHMAFields.xlsx

EventDet

rx_Aff_AuditNAHMAFields.xlsx

HHMem

rx_Aff_AuditNAHMAFields.xlsx

Income

rx_Aff_AuditNAHMAFields.xlsx

Assest

DC PSR Building Status (AFFTCPSRDC)

Top-Level Select Script Files

YSR Script Files

rs_Aff_TaxCr_PSR_DC.txt	TCPSRDC
-------------------------	---------

YSR Template Names

rx_Aff_TaxCr_PSR_DC.xlsx	TCPSRDC
--------------------------	---------

HOME Rent and Occupancy Report (AffHMRO)*Top-Level Select Script Files**YSR Script Files*

rs_Aff_HOME_RentOcc.txt	HMRNTOCC
-------------------------	----------

YSR Template Names

rx_Aff_HOME_RentOcc.xls	HMRNTOCC
-------------------------	----------

Minnesota CHART (AFFTCPSRMNCHART)*Top-Level Select Script Files**YSR Script Files*

rs_Aff_TaxCr_PSR_MN_Chart.txt	Mnchart
-------------------------------	---------

YSR Template Names

rx_aff_taxcr_psr_MN_CHART.xlsxm	Mnchart
---------------------------------	---------

AP**AP Analytics Expense Distribution (AP)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

YSR_APExpenseDistribution.xls	ExpenseD
-------------------------------	----------

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

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
<i>YSR Template Names</i>	
Select 'yDoc_KPI_LeasingActivityHistory_' + '#Period#' + '#SummarizeBy#' + '.xlsx'	ActHis
Option Expiration (OptExp)	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
<i>YSR Template Names</i>	
yDoc_Comm_OptionExpiration.xlsx	OptExp
Portfolio Summary (PortSmry)	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
<i>YSR Template Names</i>	
yDoc_KPI_PortfolioSummary.xlsx	PortSmry
Property Summary (PropSmry)	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
YSR_KPI_PropertySummary.txt	
<i>YSR Template Names</i>	
Select 'yDoc_KPI_PropertySummary_' + '#SummarizeBy#' + '.xlsx'	PropSmry
Straight-Line By Lease Summary (STLSumm)	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
<i>YSR Template Names</i>	
yDoc_Comm_StraightLineByLeaseSummary.xlsx	STLSumm
Straight-Line JE Detail (STLJEDet)	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
<i>YSR Template Names</i>	
yDoc_Comm_StraightlineRentsJEDetail.xlsx	STLJEDet
Straight-Line JE Register (STLJEReg)	
<i>Top-Level Select Script Files</i>	
<i>YSR Script Files</i>	
<i>YSR Template Names</i>	

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
------------------------------	--------

Condo

Assoc Arrears Notice Removal From PAP (AssocNoticeRemov)

Top-Level Select Script Files

rs_AssocLetter_YSR.txt	
------------------------	--

YSR Script Files

rs_AssocLetter_YSR.txt	letter
------------------------	--------

YSR Template Names

yDoc_AssocArrearsNoticeRemovalFromPAP.doc	letter
---	--------

Assoc Arrears Notice Removal From PAP Scripted (AssocNoticeSCR)

Top-Level Select Script Files

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

YSR Template Names

yDoc_AssocArrearsReminderCaveatLetter.doc

letter

Assoc Arrears Reminder3 Caveat Letter Scripted (AssocCaveatSCR)*Top-Level Select Script Files*

rs_AssocArrearsReminderCaveatLetterSCR.txt

YSR Script Files

rs_AssocArrearsReminderCaveatLetterSCR.txt

letter

YSR Template Names

yDoc_AssocArrearsReminderCaveatLetterSCR.doc

letter

Assoc Error On Cheque Letter (AssocErrorChk)*Top-Level Select Script Files*

rs_AssocErrorOnChequeLetter_YSR.txt

YSR Script Files

rs_AssocErrorOnChequeLetter_YSR.txt

letter

YSR Template Names

yDoc_AssocErrorOnChequeLetter.doc

letter

Assoc Error On Cheque Letter Scripted (AssocErrorChkSCR)*Top-Level Select Script Files*

rs_AssocErrorOnChequeLetterSCR_YSR.txt

YSR Script Files

rs_AssocErrorOnChequeLetterSCR_YSR.txt

letter

YSR Template Names

yDoc_AssocErrorOnChequeLetterSCR.doc

letter

Assoc Foreclosure Notice Letter (AssocForeclosure)*Top-Level Select Script Files*

rs_AssocLetter_YSR.txt

YSR Script Files

rs_AssocLetter_YSR.txt

letter

YSR Template Names

yDoc_AssocForeclosureNoticeLetter.doc

letter

Assoc Foreclosure Notice Letter Scripted (AssocForeclosSCR)*Top-Level Select Script Files*

rs_AssocForeclosureNoticeLetterSCR_YSR.txt

YSR Script Files

rs_AssocForeclosureNoticeLetterSCR_YSR.txt

letter

YSR Template Names

yDoc_AssocForeclosureNoticeLetterSCR.doc

letter

Assoc Welcome Condo Letter (AssocWelcomeC)*Top-Level Select Script Files*

rs_AssocWelcomeCondoLetter_YSR.txt

YSR Script Files

rs_AssocWelcomeCondoLetter_YSR.txt

letter

YSR Template Names

yDoc_AssocWelcomeCondoLetter.doc

letter

Assoc Welcome Condo Letter Scripted (AssocWelcomeCSR)*Top-Level Select Script Files*

rs_AssocWelcomeCondoLetterSCR_YSR.txt

YSR Script Files

rs_AssocWelcomeCondoLetterSCR_YSR.txt

letter

YSR Template Names

yDoc_AssocWelcomeCondoLetterSCR.doc

letter

Assoc Welcome HOA Letter (AssocWelcome)*Top-Level Select Script Files*

rs_AssocWelcomeHOALetter_YSR.txt

YSR Script Files

rs_AssocWelcomeHOALetter_YSR.txt

letter

YSR Template Names

yDoc_AssocWelcomeHOALetter.doc

letter

Assoc Welcome HOA Letter Scripted (AssocWelcomeSCR)*Top-Level Select Script Files*

rs_AssocWelcomeHOALetterSCR_YSR.txt

YSR Script Files

rs_AssocWelcomeHOALetterSCR_YSR.txt

letter

YSR Template Names

yDoc_AssocWelcomeHOALetterSCR.doc

letter

Condo Accounts Receivable Aging (CondoAR)*Top-Level Select Script Files**YSR Script Files**YSR Template Names*Select Case '#IsShowChargeCode#' When 'Yes' then 'YSR_CondoAR.xlsx' when 'No' then
'YSR_CondoAR_WithouChargeCode.xlsx' end

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

Correspondence Condo Invoice (CondoInvoice)*Top-Level Select Script Files*

rs_AssocInvoice.txt

YSR Script Files

rs_AssocInvoice.txt

Invoice

YSR Template Names

Select Case #Showaging# When 1 then 'YSR_AssocInvoice_Aging.xlsx' when 0 then

Invoice

'YSR_AssocInvoice.xlsx' end

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

Correspondence 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

Construction**Percent Complete Invoice (PCInv)***Top-Level Select Script Files**YSR Script Files*

rs_Const_YSR_PCIInvoice.txt

R1

rs_Const_YSR_PCIInvoice_Detail.txt

R2

YSR Template Names

rs_Const_YSR_PCIInvoice.xlsx

R1

rs_Const_YSR_PCIInvoice_Detail.xlsx

R2

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

CRM**CRM Analytics - Consolidated Report (CRMAAnaly)***Top-Level Select Script Files**YSR Script Files**YSR Template Names*

```
SELECT 'yDoc_CRMAnalytics' + (CASE '#ShowUnassignedLeads#' WHEN 'Yes' THEN CASE
 '#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
```

CRMAAnaly

```
'#SummaryType#' WHEN 'Property' THEN '_Prop' WHEN 'Leasing Agent' THEN '_LA' WHEN
'Customer' THEN '_Cust' END END+'.xlsx'
```

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

<i>YSR Template Names</i>		
rs_YSR_CreditNote.docx		credit
Custom Financial Income Statement (CF_IS)		
<i>Top-Level Select Script Files</i>		
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
Income Statement (IncomeSt)		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
YSR_IncomeStatement.xlsx		IncomeSt
Invoice Register (InvoiceR)		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
YSR_TransRegister_Invoice.xlsx		InvoiceR
Payable Register (PayableR)		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
YSR_TransRegister_Payable.xlsx		PayableR
Receipt Register (Receipt)		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		
<i>YSR Template Names</i>		
YSR_TransRegister_Receipt.xlsx		Receipt
Trial Balance (TrialBal)		
<i>Top-Level Select Script Files</i>		
<i>YSR Script Files</i>		

YSR Template Names

YSR_TrialBalance.xlsx

TrialBal

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

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 (IntResMktRentSch)*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

PRDL

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

InternationalEU**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

Investment Management

Capital Balance Report Investment View (cbal_inv)*Top-Level Select Script Files**YSR Script Files**YSR Template Names*

capbal_investment.xlsx

capbal11

Capital Balance Report Investor View (cap_bal)*Top-Level Select Script Files**YSR Script Files**YSR Template Names*

capbal_data.xlsx

capbal

Capital Call Report (capcallre)*Top-Level Select Script Files*

rs_capcall_report_header.txt

YSR Script Files

rs_capcall_report_detail.txt

capcall

YSR Template Names

rs_capitalcall_report.xlsx

capcall

Capital Call Report Single Currency (singcpcl)*Top-Level Select Script Files*

rs_capcall_singlecurr_report_header.txt

YSR Script Files

rs_capcall_singlecurr_report_detail.txt

capcall

YSR Template Names

rs_capitalcall_singlecurr_report.xlsx

capcall

Custom IM Investment Attribute Report (custIM_invsatt)*Top-Level Select Script Files*

*YSR Script Files**YSR Template Names*

custim_invstmntattr.xlsx

cusim

Custom IM Investment Holding Report (CustIM_Invstmnt)*Top-Level Select Script Files**YSR Script Files**YSR Template Names*

customim_investment.xlsx

custim

Custom IM Investor Attribute Report (cusIM_invattr)*Top-Level Select Script Files**YSR Script Files**YSR Template Names*

customim_investorattr.xlsx

custIM

Custom IM Investor Holding Report (CustIM_invstr)*Top-Level Select Script Files**YSR Script Files**YSR Template Names*

customim_investor.xlsx

custIM

Distribution Notice (dstnotic)*Top-Level Select Script Files*

rs_distnotice_header.txt

YSR Script Files

rs_distnotice_detail.txt

dstnot

YSR Template Names

distnotice.xlsx

dstnot

ILPA Capital Call (ILPACap)*Top-Level Select Script Files*

SS_ILPA_capitalcall_header.txt

YSR Script Files

SS_ILPA_capitalcall_detail.txt

Report

YSR Template Names

SS_ILPA_Capitalcall.xlsx

Report

ILPA Fund Report (FundRpt)*Top-Level Select Script Files*

SS_YSR_ILPA_Script.txt

YSR Script Files

SS_YSR_ILPA_Script.txt

ILPA

SS_YSR_ILPA_Script.txt

ExSumm

YSR Template Names

YSR_ILPA_Page_BalanceSheet.xlsx

ILPA

YSR_ILPA_CapitalAccountStatement.xlsx

CAS

YSR_ILPA_Page_21.xlsx

ExSumm

Pref_Deal_Listing (PR_DLList)

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 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

Legal**Legal Collections (LegalCollections)***Top-Level Select Script Files**YSR Script Files*

YSR_Legal_Collections.txt

LglColl

YSR Template Names

YSR_Legal_Collections.xlsx

LglColl

Legal Custom Notice in YSR (LegalCustmNotice)*Top-Level Select Script Files*

rs_Legal_Custom_Notice_YSR.txt

YSR Script Files

rs_Legal_Custom_Notice_YSR.txt

LglNotic

YSR Template Names

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)*Top-Level Select Script Files*

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'

LglNotic

when 'Public Housing' then 'YSR_LegalNoticeNewVersion_PHA.xlsx' Else

'YSR_LegalNoticeNewVersion_RES.xlsx' end

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

PHA

14.2 FSS Participants Escrow (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 'rx_PHA_YSR_FSS_Participants_Escrow_Detail.xlsx'

FSSE

ELSE 'rx_PHA_YSR_FSS_Participants_Escrow_Summary.xlsx' END

Active Units (ActiveUnits)

Top-Level Select Script Files

YSR Script Files

rs_PHA_Active_Units_YSR.txt

ActDet

SR Template Names

rx_PHA_Active_Units_YSR.txt

ActDet

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

BVCDL

rs_PHA_YSR_Census_Demographics.txt

BVCD

rs_PHA_YSR_Census_Demographics.txt

BVCDG

YSR Template Names

rx_PHA_YSR_Census_Demographics_Listing.xlsx

BVCDL

rx_PHA_YSR_Census_Demographics_CrossTab.xlsx

BVCD

rx_PHA_YSR_Census_Demographics_Graphical.xlsx

BVCDG

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 'No' GLByMon
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 Script Files

rs_PHA_YSR_HUDForecasting.xlsx

Forecast

Income Listing- Single Income Type (Inclstng)*Top-Level Select Script Files*

rs_PHA_YSR_IncomeList.txt

YSR Script Files

rs_PHA_YSR_IncomeList.txt

Inclstng

rs_PHA_YSR_IncomeList.txt

IncTot

YSR Template Names

```
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
rx_PHA_YSR_IncomeList_With_GrandTotal.xlsx
```

Inclstng

IncTot

Inspection Checklist - HUD 52580 (IChkLst)*Top-Level Select Script Files*

rs_PHA_YSR_Voucher_Inspection_Checklist_HUD52580.txt

YSR Script Files

rs_PHA_YSR_Voucher_Inspection_Checklist_HUD52580.txt

IChkLst

YSR Template Names

rx_PHA_YSR_Voucher_Inspection_Checklist_HUD52580.xlsx

IChkLst

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_DefOptList.xlsx

basic

PHA Demographics (PHADemo)*Top-Level Select Script Files**YSR Script Files*

rs_PHA_YSR_Demographics.txt

PHADemo

YSR Template Names

rs_PHA_YSR_Demographics.xlsx

PHADemo

PHA Recert Outcome Report (RecertOutcome)*Top-Level Select Script Files**YSR Script Files*

rs_PHA_YSR_RecertOutcome.txt

basic

YSR Template Names

rx_PHA_YSR_RecertOutcome.xlsx

basic

PHA Setup Review (PHASetup)*Top-Level Select Script Files**YSR Script Files*

rs_PHA_YSR_Analysis.txt

PropSet

YSR Template Names

rx_PHA_YSR_Analysis.xlsx

PropSet

Portability Billing Statement (PortBill)*Top-Level Select Script Files*

rs_PHA_YSR_Portability_Statement.txt

YSR Script Files

rs_PHA_YSR_Portability_Statement.txt

PortBill

rs_PHA_YSR_Portability_Statement.txt

Transact

YSR Template Names

rx_PHA_YSR_Portability_Statement.xlsx

PortBill

rx_PHA_YSR_Portability_Statement_Transact.xlsx

Transact

Post Subsidies Transaction Recap Report (PRHTranR)*Top-Level Select Script Files*

rs_PHA_YSR_PRH_TransactionRecap.txt

YSR Script Files

rs_PHA_YSR_PRH_TransactionRecap.txt

TrRec

rs_PHA_YSR_PRH_TransactionRecap.txt

TRPort

rs_PHA_YSR_PRH_TransactionRecap.txt

TRC

rs_PHA_YSR_PRH_TransactionRecap.txt	HAP
rs_PHA_YSR_PRH_TransactionRecap.txt	URP
rs_PHA_YSR_PRH_TransactionRecap.txt	PRT
rs_PHA_YSR_PRH_TransactionRecap.txt	PAF
rs_PHA_YSR_PRH_TransactionRecap.txt	POH
rs_PHA_YSR_PRH_TransactionRecap.txt	POA
rs_PHA_YSR_PRH_TransactionRecap.txt	HTH
rs_PHA_YSR_PRH_TransactionRecap.txt	OTF
rs_PHA_YSR_PRH_TransactionRecap.txt	FSS
rs_PHA_YSR_PRH_TransactionRecap.txt	TPA

YSR Template Names

rx_PHA_YSR_PRH_TransactionRecap.xlsx	TrRec
rx_PHA_YSR_PRH_TransactionRecapPort.xlsx	TRPort
rx_PHA_YSR_PRH_TransactionRecap_TRCDet.xlsx	TRC
rx_PHA_YSR_PRH_TransactionRecap_HAPDet.xlsx	HAP
rx_PHA_YSR_PRH_TransactionRecap_URPDet.xlsx	URP
rx_PHA_YSR_PRH_TransactionRecap_PRTDet.xlsx	PRT
rx_PHA_YSR_PRH_TransactionRecap_PAFDet.xlsx	PAF
rx_PHA_YSR_PRH_TransactionRecap_POHDet.xlsx	POH
rx_PHA_YSR_PRH_TransactionRecap_POADet.xlsx	POA
rx_PHA_YSR_PRH_TransactionRecap_HTHDet.xlsx	HTH
rx_PHA_YSR_PRH_TransactionRecap_OTFDet.xlsx	OTF
rx_PHA_YSR_PRH_TransactionRecap_FSSDet.xlsx	FSS
rx_PHA_YSR_PRH_TransactionRecap_TPADet.xlsx	TPA

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
-----------------------------	-----

Rent Comparability Subject Unit Report (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 Unit Setup Review (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 Unit Update Letter (RCUnitU)*Top-Level Select Script Files*

rs_PHA_YSR_RC_Unit_Update_New.txt

YSR Script Files

rs_PHA_YSR_RC_Unit_Update_New.txt

RCUnitU

YSR Template Names

rx_PHA_YSR_RC_Unit_Update_New.xlsx

RCUnitU

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 'rx_PHA_YSR_Recerts_Annually_CrossTab.xlsx' END AnReExam

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
          VerType
    '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
  
```

VMS - GL Reconciliation Report (VMSG)*Top-Level Select Script Files**YSR Script Files*

rs_PHA_YSR_VMS_GL_Reconciliation_Report.txt

VMSG

YSR Template Names

rx_PHA_YSR_VMS_GL_Reconciliation_Report.xlsx

VMSG

VMS Comparison Report (VMSC)*Top-Level Select Script Files**YSR Script Files*

rs_PHA_YSR_VMS_Comparison.txt

VMSC

YSR Template Names

rx_PHA_YSR_VMS_Comparison.xlsx

VMSC

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

VMS Summary (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

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.xlsxVou_Out

Waiting List Family History (WLFamilyHist)

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'	WL
ELSE 'rx_WL_YSR_Waiting_List_FamHist_Position.xlsx' END	

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
--------------------------	----

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) (CompTranNotice)

Top-Level Select Script Files

rs_ITF_RI_Compliance_Transfer_Notice_Filter.txt

YSR Script Files

rs_ITF_RI_Compliance_Transfer_Notice_Script.txt

ExprdPol

YSR Template Names

yDoc_Renters_Insurance_Compliance_Transfer_Notice_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

resactvrt

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

unitavlbl

Senior Housing

Assessment & Service Result Report (EHRAssSerRes)

Top-Level Select Script Files

YSR Script Files

rs_SeniorEHRXAssessmentResultServiceNew.txt

SerRep

YSR Template Names

rs_SeniorEHRXAssessmentResultServiceNew.xlsx

SerRep

Assessment Result Report (EHRAssRes)

Top-Level Select Script Files

YSR Script Files

rs_SeniorEHRXAssessmentResultNew.txt

AssRep

YSR Template Names

rs_SeniorEHRXAssessmentResultNew.xlsx

AssRep

Daily Task Log (EHRXDTL)

Top-Level Select Script Files

YSR Script Files

rs_SeniorEHRXDailyTeamAssignment.txt

SDLR

YSR Template Names

rs_SeniorEHRXDailyTeamAssignment.xlsx

SDLR

Monthly Task Log (EHRMTLLink)

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 (EHRMTL)

Top-Level Select Script Files

rs_SeniorEHRXMonthlyTaskLogYSR.txt

YSR Script Files

rs_SeniorEHRXMonthlyTaskLog.txt

MTI

YSR Template Names

rs_SeniorEHRXMonthlyTaskLog.xlsx

MTI

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 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

Senior Aging by Accounting Period (SeniorAging)*Top-Level Select Script Files**YSR Script Files*

rs_SeniorAnalytic.txt

Com

rs_SeniorAnalytic.txt

Res

rs_SeniorAnalytic.txt

Pay

rs_SeniorAnalytic.txt

Chr

YSR Template Names

SeniorAging.xlsx

Com

SeniorAgingResident.xlsx

Res

SeniorAgingPayer.xlsx

Pay

SeniorAgingChargeCode.xlsx

Chr

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**YSR Script Files*

rs_SeniorAnalytic.txt

TrdActWk

rs_SeniorAnalytic.txt

TrdActMt

YSR Template Names

SeniorTrendActivityWeekly.xlsx

TrdActWk

SeniorTrendActivityMonthly.xlsx

TrdActMt

Senior Trend Occupancy (SeniorTrendOcc)*Top-Level Select Script Files**YSR Script Files*

rs_SeniorAnalytic.txt

TrOccW

rs_SeniorAnalytic.txt

TrOccM

YSR Template Names

SeniorTrendOccupancyWeekly.xlsx

TrOccW

SeniorTrendOccupancyMonthly.xlsx

TrOccM

Senior Unit Availability (SeniorUnitAvai)*Top-Level Select Script Files**YSR Script Files*

rs_SeniorAnalytic.txt

UnitAvai

YSR Template Names

SeniorUnitAvailability.xlsx

UnitAvai

Senior Weekly Conversion Ratio (SeniorWeekConv)*Top-Level Select Script Files**YSR Script Files*

rs_SeniorAnalytic.txt

WkConv

YSR Template Names

SeniorWeeklyConversion.xlsx

WkConv

Tax and Insurance**TI Commercial Insurance Policies (TICINSPOL)***Top-Level Select Script Files*

rs_TI_CommlInsurancePolicies.txt

YSR Script Files

rs_TI_CommlInsurancePolicies.txt

INSUDET

YSR Template Names

ydoc_TI_CommlInsurancePolicies.xlsx

INSUDET

TI Insurance Claims (INSCLAIM)*Top-Level Select Script Files*

rs_TI_InsuranceClaims.txt

YSR Script Files

rs_TI_InsuranceClaims.txt

INSCLAIM

YSR Template Names

ydoc_TI_InsuranceClaims.xlsx

INSCLAIM

TI Insurance Coverages (TIINSCOV)*Top-Level Select Script Files*

rs_TI_InsuranceCovrances.txt

YSR Script Files

rs_TI_InsuranceCovrances.txt

INSUDET

YSR Template Names

ydoc_TI_InsuranceCoverages.xlsx

INSUDET

TI Insurance Relationships (TIINSREL)*Top-Level Select Script Files*

rs_TI_Relationships.txt

YSR Script Files

rs_TI_Relationships.txt

INSREL

YSR Template Names

ydoc_TI_Relationships.xlsx

INSREL

TI Missing Insurance Policies (TIMISINS)*Top-Level Select Script Files*

rs_TI_MissingInsurance.txt

YSR Script Files

rs_TI_MissingInsurance.txt

MISSINS

YSR Template Names

ydoc_TI_MissingInsurance.xlsx

MISSINS

TI Tax Appeals (TITAXAPP)*Top-Level Select Script Files*

rs_TI_TaxAppeals.txt

YSR Script Files

rs_TI_TaxAppeals.txt

TAXAPP

YSR Template Names

ydoc_TI_TaxAppeals.xlsx

TAXAPP

TI Tax Assessments (TIASMVAL)*Top-Level Select Script Files*

rs_TI_TaxAssessments.txt

YSR Script Files

rs_TI_TaxAssessments.txt

TAXASMTD

YSR Template Names

ydoc_TI_TaxAssessments.xlsx

TAXASMTD

TI Tax Assessments (TITAXASM)*Top-Level Select Script Files*

rs_TI_TaxAssessments.txt

YSR Script Files

rs_TI_TaxAssessments.txt	TAXASMTD
rs_TI_TaxParcels.txt	TAXPAR

YSR Template Names

ydoc_TI_TaxParcels.xlsx	TAXPAR
ydoc_TI_TaxAssessments.xlsx	TAXASMTD

TI Tax Exemptions (TITAXEXE)*Top-Level Select Script Files*

rs_TI_TaxExemptions.txt

YSR Script Files

rs_TI_TaxExemptions.txt	TAXEXE
-------------------------	--------

YSR Template Names

ydoc_TI_TaxExemptions.xlsx	TAXEXE
----------------------------	--------

TI Tax Parcels (TITAXPAR)*Top-Level Select Script Files*

rs_TI_TaxParcels.txt

YSR Script Files

rs_TI_TaxParcels.txt	TAXPAR
----------------------	--------

YSR Template Names

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
-------------------------------	--------

Utility Billing

Utility Billing All Meters (UBMeterSetupAll)*Top-Level Select Script Files**YSR Script Files*

rs_YSR.Utility.Billing.Meter.Setup.All_Report.txt UBMeters

YSR Template Names

YSR.Utility.Billing.Meter.Setup.All_Report.xlsx UBMeters

Utility Billing Invoice (UB_Invoice)

Top-Level Select Script Files

rs_YSR.InternationalMEAA.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

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_UtilityBillingAnnualReviewSummary.xlsx	propSum
rs_UB_UtilityBillingAnnualReview.xlsx	PropDet

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
-------------	--------

APPENDIX A

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.

AP Analytics

AP Analytics Expense Distribution (AP)

AR Analytics

Aging Detail Report (AginDet)

AR Aging Summary Report (AgingSum)

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)

Portfolio Summary (PortSmry)

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)

Condo Correspondence Invoice

Correpondence Condo Invoice (CondoInvoice)

Condo Correspondence Ledger

Correpondence 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)

Correpondence Condo Letter (CndLetter)

Condo Correspondence Purchase

Correpondence 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 (CMIInv)

CRM

CRM Analytics - Consolidated Report (CRMAAnaly)

Custom Financials

Custom Financial Income Statement (CF_IS)

Custom IM Data

Custom IM Investment Attribute Report (custIM_invsatt)

Custom IM Investment Holding Report (CustIM_Invstmnt)

Custom IM Investor Attribute Report (cusIM_invattr)

Custom IM Investor Holding Report (CustIM_invstr)

Financial Analytics

Balance Sheet (BalanceS)

Budget Comparison (BudgetCo)

Income Statement (IncomeSt)

Trial Balance (TrialBal)

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)

IM Custom Financial Data

ILPA Fund Report (FundRpt)

IM Data

ILPA Fund Report (FundRpt)

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)

Payable Register (PayableR)

Receipt Register (Receipt)

PR Preferred Returns Analytics

Pref_Deal_Listing (PR_DList)

Pref_Equity_Multiple (PR_EqMul)

Pref_Hypothetical_Adjustment (PR_HypCa)

Pref_IRR (PR_IRR)

Pref_PerfCalculationSummary_PeriodAsColumn (PR_PCSPC)

Pref_Periodic_Calculation_Summary (PR_PCSum)

Pref_PeriodicRuleSetup (PR_PRS)

Pref_SpecialDistribution (PR_SDist)

Pref_SpecialRuleSetup (PR_SDRS)

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)

Tenancy Schedule

Tenancy Schedule (Tenancy)

APPENDIX A

Constant Values for Analytic 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 fields for 'Merged Report' (set to 'Aging Detail Report (AginDet)'), 'Report Code' (set to 'AgingDet'), and 'SectionCode' (set to 'Aging'). Below these are 'Save' and 'Close' buttons. To the left of the main table, a vertical callout labeled 'A/R Analytics filter elements' points to the first column of the table. The table has three columns: 'Standard Report Filter', 'Report Filters', and 'Constant Value'. The data is as follows:

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
Save	Close

Standard Report Filter

PerfTable	ASCAP_AssetInfo
PerfField	dCashBalance^dForecastGain^dFo
ReportStyle	investmentview
PropertyCode	Investor
FromMMYY	FromDate
ToMMYY	ToDate

Report Filters

Cash Balance	&=fundperf.dCashBalance
Forecast Gain	&=fundperf.dForecastGain
Forecast IRR	&=fundperf.dForecastIRR
Share Price	&=fundperf.dSharePrice
Market Value	&=fundperf.A_BEGINNINGMARKETVALUE

Constant Value

Fund Performance Data :

Cash Balance	&=fundperf.dCashBalance
Forecast Gain	&=fundperf.dForecastGain
Forecast IRR	&=fundperf.dForecastIRR
Share Price	&=fundperf.dSharePrice
Market Value	&=fundperf.A_BEGINNINGMARKETVALUE

AP Analytics

ProgramType	YES
ReportType	Aging
ReportType	Expense Distribution
ReportType	Vendor Ledger
ShowDetail	YES

AR Analytics

ChargeCode	SELECT " -- #hproperty#
ChargeCode	SELECT " --'##'
ChargeCode	SELECT ct.hmy from chargtyp ct inner join lookup l on l.sdesc = ct.scode where l.listname = N'reschargecode' /* # # */
ChargeCode	SELECT dbo.Select_CommaSeperatedChgCodes()
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
MinAmount	1
NoOfSlabs	1
NoOfSlabs	2
NoOfSlabs	3
NoOfSlabs	4
NoOfSlabs	5
NoOfSlabs	6
OwnerId	0
OwnerId	SELECT " -- #hproperty#
OwnerId	SELECT hmyperson from owner
ReportType	Aging Detail
ReportType	Aging Summary
ReportType	Receivable Detail
RunToday	1
SlabDays	30
SlabDays	300
SlabDays	45
SlabDays	60
SummarizeBy	Lease by Charge Code
SummarizeBy	Lease/Payee

SummarizeBy	Property
SummarizeBy	Property by Charge Code

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	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	RentRoll
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

Correspondence (International)

ChargeType	All
ChargeType	CAM
ChargeType	MISC
ChargeType	OVG
ChargeType	RENT
IsReminderTypeInvoiceNumber	True
ReportType	11
ReportType	12
ReportType	2
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
ShowUnassignedLeads	No
ShowUnassignedLeads	Yes
SummaryType	Customer
SummaryType	Leasing Agent
SummaryType	Property

Custom Financials

AttributeSortName	Consolidation Status
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	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
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
IsConsolidate	1
IsConsolidate	FALSE
IsConsolidate	No

IsConsolidate	TRUE
IsConsolidate	Yes
PropertyId	-1
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 '#bSuppressZero#' When 'Yes' then 'TRUE' Else 'False' End
SuppressZero	SELECT case when '#SupressZero#' = 'Yes' then 1 else 0 end
SuppressZero	True
SuppressZero	Yes
TreeLevel	1
TreeLevel	2
TreeLevel	3
TreeLevel	4

Custom IM Data

AttributeName	SELECT sname TextField, Sname ValueField from AttributeName
DateInUse	Effective Date
DateInUse	Post Month
DateInUse	Tran Date
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

Financial Analytics

DecimalDigits	0
DecimalDigits	1
FinType	BALANCE SHEET
FinType	Budget Comparison
FinType	Cash Flow
FinType	Income Statement
FinType	TRIAL BALANCE
Graph	0
Grid	1
IncludeNotes	1
IsConsolidate	0
IsConsolidate	1
PropertyId	-1
SuppressZero	0
SuppressZero	1
SuppressZero	SELECT case when '#SuppressZero#' = 'Yes' then 1 else 0 end
SuppressZero	True
TreeLevel	1
TreeLevel	10
TreeLevel	2

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.CUSTOM_GetProductParams(N'#ProductCo de#',N'dynamiclist',N'',0)
InvestmentCodes	SELECT

```

Dbo.CBREGIUS_GetProductParams(N'#ProductCode#',N'dynamiclist',
N'',0) -- #PropertyCode#
InvestorCodes
SELECT
Dbo.CUSTOM_GetProductParams(N'#ProductCo
de#',N'entity',N'',0)
InvestorCodes
SELECT
Dbo.CBREGIUS_GetProductParams(N'#ProductCode#',N'entity',N'',0) -
- #PropertyCode#
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
NumberFormat
1
NumberFormat
100
NumberFormat
1K
NumberFormat
1M
ReportingCurrency
Investment
ReportingCurrency
Investor
ReportStyle
Investment
ReportStyle
Investor
ReportType
Capital Roll Forward
ShowOwnershipColumn
-1

```

IM Custom Financial Data

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.CUSTOM_GetProductParams(N'#ProductCo de#,N'entity',N'',0)
InvestorCodes	SELECT dbo.CUSTOM_InvestmentCodes(N'Joint Venture (Asset Level)', N'#ProductCode#')
InvestorCodes	SELECT dbo.CUSTOM_Propfilter('#InvestorID#')
InvestorCodes	SELECT dbo.CBREGI_InvestmentCodes(N'Joint Venture (Asset Level)', N'#ProductCode#')
InvestorCodes	SELECT
	Dbo.CBREGIUS_GetProductParams(N'#ProductCode#,N'entity',N'',0)
InvestorCodes	SELECT dbo.CPR20619_Propfilter('#InvestorID#')
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	-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

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 Transaction Registers

SortOn	Person
TranDetail	1
TranDetail	SELECT Case '#ReportType#' When 'Charge' Then NULL Else 1 eND
TranType	Charge
TranType	Invoice
TranType	Payable
TranType	Receipt

PR Preferred Returns Analytics

ActivityType	Accrual Only
ActivityType	Both
ActivityType	Distribution Only
ReportType	Deal Listing
ReportType	Hypothetical Adjustment Calculation
ReportType	IRR Hurdle Report
ReportType	Periodic Calculation Rules Setup
ReportType	Periodic Calculation Summary
ReportType	Special Distribution
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
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 MyOrderBy ORDER BY MyOrderBy</pre>
Summarizeby	<pre>SELECT 'Community' TextField, 'ByCommunity' ValueField,1 MyOrderBy UNION SELECT 'Market Area' Text, 'ByMaketArea' 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</pre>

```

MyOrderBy UNION SELECT 'Sales Counselor' Text, 'BySalesCounselor'
Value,2 MyOrderBy UNION SELECT 'Care Level' Text, 'ByCareLevel'
Value,3 MyOrderBy ORDER BY MyOrderBy
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
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

```

Tenancy Schedule

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
valuationtype	SELECT sCode TextField, valtype ValueField FROM COMMVALUATIONTYPE

Index

A

ABF Valuations, 54
 aliases, 14, 25, 28, 33, 91
 analytics report, 14, 15, 27, 65, 90
 attachments, 2, 4, 24, 43, 45, 88, 100

B

batch number, 54
 budget, 54

C

caution symbol, x
 class name, 100, 103
 column names, 28
 contacts, 43, 44, 46
 control number, 54
 custom correspondence, 11
 custom filter fields, 11, 49, 50, 54, 70, 71, 72, 192
 Custom Financial Analytics, 110, 112
 custom side menu
 adding, 106, 107
 custom token, 11, 33, 57, 91
 CustomCorrespMerging, 11, 16, 201

D

data
 multiple rows, 14, 86
 data column, 2, 11, 12, 23, 25, 57
 data definition language, 39
 data manipulation language, 39
 deliverables, 16
 details, 163
 drill-down, 172

E

email, 43, 46, 47
 email ID, 43
 email template, 46
 email type, 46
 Excel, ix, 147, 148, 150, 161, 168
 hyperlinks, adding, 172

F

filter condition, 11, 59, 85
 filter mapping, 11, 22, 49, 50, 65, 71, 74
 filtration, 22, 34, 49, 52, 65, 72, 75
 foreign key, 11, 12, 23
 formatting
 grouping, 153
 labeling, 157
 subtotaling, 153
 formula smart markers, 148, 151
 formulas
 dynamic, 151, 152
 repeating dynamic, 151, 152
 freeze pane, 171, 172

G

generating reports, 98, 174
 graphs, 1, 2, 110
 on screen, 7, 20
 grouping data, 153

H

Help, x
 Help buttons, x
 html, 94
 hyperlinks, 172

I

images, 93, 147, 161, 162
 information symbol, x
 Investor Portal, 2, 4
 iTypes, 193

K

key column, 12, 15, 21, 23, 25, 46, 198

L

list
 dynamic, 60
 values, 56

lookup list
 custom, 57, 60
 values, 56
 YSI, 51, 52, 53, 56

M

merge field, 12, 13, 15, 90, 92
 merged reports, 3, 10, 12

N

notes
 caution, x
 information, x
 tip, x

O

online Help, x

P

p.hmy in (#property#), 15, 34, 53, 70, 71, 74, 190, 194, 199
 page break column, 3, 10, 12, 22
 parameters
 grouping, 153
 parent-child, 54
 picture, 161, 162, 163
 pivot table, 1, 86, 110
 on screen, 8, 20
 primary key, 11, 12, 23, 25, 26, 190, 198
 program object iTypes, 193

R

relational key, 12
 report scheduler, 104
 report
 section, 3, 10, 11, 12, 13, 14, 16, 34, 50, 58, 68, 85, 183
 report-generation
 screen, 10, 11, 13, 15, 21, 24, 26, 29, 47, 56, 57, 60, 100, 107, 192, 198, 199
 Reports

path, 15, 17, 20, 22, 23, 27, 95, 118, 123, 176, 189, 197
 roles, 43, 44

S

scroll bars, 171
 section code, 13, 37, 148, 183, 184
 select section name, 183
 SharePoint, 2, 4, 46, 47, 87, 100
 smart markers, 14, 183
 basic, 148
 formula, 148, 151
 image, 161
 overview, 147
 parameters, 148, 149, 150, 185
 sReturnType, 31
 standard report, 15, 28, 31, 65, 68, 85, 120, 125

T

t.hmyperson in (#tenant#), 71, 72, 74
 task runner, 2, 100, 103
 class name, 100, 103
 parameters, 100
 url, 100
 templates, 3, 4, 13, 15, 16, 22, 27, 35, 116, 175, 176, 183, 186
 Custom Financial Analytics, 16, 110
 email, 45, 46
 error checking, 186
 Excel, 186
 temporary tables, 39
 tenant, 23, 43, 46, 54, 57, 70, 71, 72, 121, 184, 194
 tip symbol, x
 tokens, 33, 49, 69, 70, 85, 190, 191, 194, 199, 200
 invalid, 92
 named, 69, 70, 72, 75, 194
 numbered, 69, 70, 71, 73
 session and user-related, 38
 top-level select
 statement, 15, 20, 23, 24, 25, 37, 50, 191, 198

U

unique identifier, 13, 15, 21, 22, 23, 46, 62, 198

V

Val 1
 parameter, iii, 34, 53, 69, 70, 71, 72, 73, 74, 75, 76, 194
 Voyager
 analytics, 1, 2, 11, 13, 15, 18, 22, 27, 28, 34, 49, 50, 57, 64, 8

5, 91, 174, 175, 178, 182
discovering field aliases, 91

W

Word
template setup example, 80

Y

Yardi Excel Add-In, x, 174, 175, 189, 190
filter section, 193
ysiPropertyListLookup, 51, 52, 53
ysiPropertyLookup, 51, 52
ysiPropertyOrDCFPropertyLookup, 51, 52
ysiPropertyOrListLookup, 51, 52
YSR report
filter, 3, 11, 12, 16, 18, 19, 22, 34, 49, 50, 51, 54, 57, 73, 75
YSR Smart Marker Designer, 174, 177, 183
YSROptions, 163