



Sharperlight 4.6

Statement of Release

www.sharperlight.com

info@sharperlight.com

Sharperligh 4.6 Statement of Release

Published by Philight Software International Pty Ltd

Copyright 2008-2017 Philight Software International Pty Ltd

All other copyrights and trademarks are the property of their respective owners

Printed: August 2017

Document Version: 1.0.4

Disclaimer: The information in this document remains the current view of Philight Software International Pty Ltd and is subject to change without notice. This position is due to changing market conditions and should not be interpreted as a commitment to the correct operation of any technology or product contained herein.

This document is intended as information only and Philight Software International Pty Ltd makes no warranties, express or implied as to the information in this document.

All rights reserved.

The copyright of this document and the computer software described herein and provided herewith are the property of Philight Software International Pty Ltd. No part of this publication or the computer software may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means or otherwise used without the express written permission of Philight Software International Pty Ltd.



Philight Software International Pty Ltd
15 Ohio Place
Marangaroo
WA 6065
AUSTRALIA

Table of Contents

Part I Introduction	4
1 Release History	4
Part II Key Functionality	5
1 Migration to .NET Framework 4.6.1	5
2 New Installer	5
3 HTML5 Data Entry Grid	6
Part III Enhancements	7
1 Excel Add-in	7
Writeback Options	7
2 Query Builder	8
Faster Expression Engine	8
Wildcard Search	8
Drill To	9
3 Web Channel	9
Auto-search and Multi-select	10
Browser Resource Caching	11
4 Foundry	11
Auditing	11
Templates	12
State Values	12
5 Studio	12
FoxPro Connection Template	12
ODBC DSN Connection Templates	12
6 Site Setup	13
Remote Access	13
7 Other	13
Display Scaling	13
File Associations	14
Backup	14

1 Introduction

This document outlines the August 2017 release of Sharperligh 4.6. There is a big jump in the minor version number from the 4.1 release in March 2017 and this is to highlight the key deliverable, that Sharperligh now relies on the 4.6.1 .NET Framework. This release has an internal number of 4.6.155 for the 64 bit build and 4.6.156 for the 32 bit build.

The migration of the code-base from .NET 3.5 to 4.6 has necessitated some code changes but it has provided the opportunity to re-examine and optimise functionality. Changing the .NET runtime has forced the need for a new application packaging and installer interface, and the new installer now allows selective installations with client, server and development.

Three months were set aside for the migration to .NET 4.6.1 and the testing regime, fortunately it did not take nearly that long to stabilise the necessary changes and time was available to focus on new development. With this release, there is a new HTML5 Data Entry Grid which now replaces the deprecated Silverlight grid. There are performance improvements, most noticeably with the caching of web content from the Sharperligh Service and the Expression engine runs 40% faster. Generally there are a number of enhancements base on partner feedback and the experience of consultants on current projects.

As this release illustrates, feedback from software partners and customers is extremely valuable because it adds real world experience and it prioritises our development efforts. We welcome suggestions from our customers and encourage them to send there insights to support@sharperligh.com.

1.1 Release History

The Sharperligh development team uses a continuous build process where minor builds are made available to consultants, software partners and early adopters; to beta test and provide feedback. Statements of intent are drafted and distributed several months after each release and they provide a roadmap for each successive major release.

Major Build Numbers	Release Date
2.6.0	February 2012
2.7.4	September 2012
2.8.22	March 2013
2.9.8 / 2.9.7 32 bit	October 2013
2.10.3 / 2.10.4 32 bit	May 2014
3.0.5 / 3.0.6 32 bit	November 2014
3.1.19 / 3.1.20 32 bit	October 2015
4.0.18 / 4.0.19 32 bit	October 2016
4.1.14 / 4.1.15 32 bit	March 2017
4.6.155 / 4.6.156 32 bit	August 2017

Each major release has several key pieces of new functionality that are the cornerstone of the release. The table below is a list of these key features.

Major Releases	Key Functionality
2.6	Multilingual, Query Union
2.7	Instance Management, HTML Charts
2.8	Dashboards, Excel Ribbon
2.9	Materialised Query, Favourites
2.10	Application Licensing, Page Designer
3.0	Report Rows, Web Licensing
3.1	System Data Sources, Excel Splits, Excel Sheet Formula
4.0	Scheduler, Foundry, Active Data Entry
4.1	Query In-memory Cache
4.6	.NET Framework 4.6.1, HTML5 Data Entry Grid

2 Key Functionality

2.1 Migration to .NET Framework 4.6.1

Sharperlight has been relying on .NET 3.5 for many years but the time has now come to migrate the code-base to .NET Framework 4.6.1. This is a conservative decision because 4.6.1 was released in November 2015 and it is well known in the software industry. This new framework has significant improvements in communication, presentation, networking and connectivity. It also includes new APIs that can be used in integrating the Sharperlight query engine with cloud data sources.

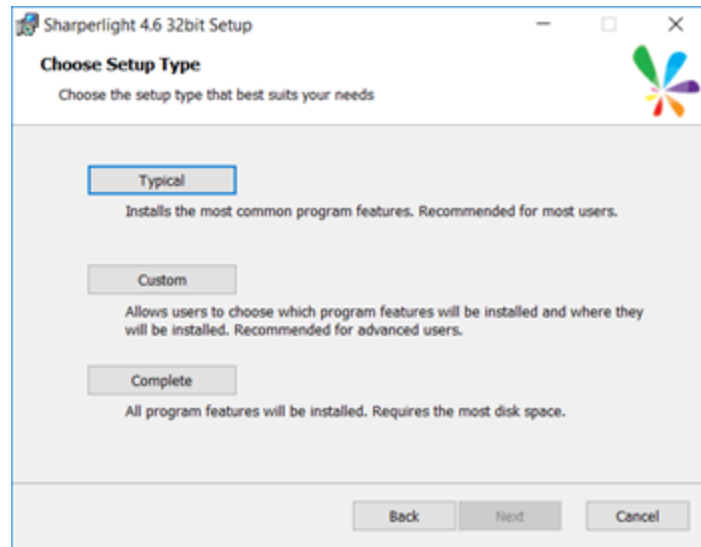
The migration of the code from .NET 3.5 to .NET 4.6.1 was relatively easy but intensive testing has been completed to ensure that like for life functionality has been maintained between releases. The move to .NET Framework 4.6.1 has changed the supported operating systems with Windows 7 Service Pack 1 and Windows Server 2008 R2 becoming the minimum operating requirements. Sharperlight 4.1 will be the last release that supports Windows XP, Windows Vista and Windows Server 2008.

2.2 New Installer

This release heralds in a new Setup Wizard courtesy of the WiX toolset for building Windows installation packages. There are now two installers with separate installation files for 32 bit and 64 bit Windows operating systems. The installers behave the same but the 32 bit package will only operate in 32 bit mode, irrespective of the destination operating system. Obviously, the 64 bit installer cannot be installed on a 32 bit Windows environment.

The new installer cannot automatically uninstall an existing Sharperlight 4.1 or earlier installation. Therefore, all previous installations of Sharperlight will need to be manually uninstalled before installing 4.6 or later. The configuration settings in Client Setup and existing datamodels (if any) are not removed during an uninstall and they will be reused by the new installer. The logon details and startup type for the Sharperlight Service will be lost in a 4.1 or earlier uninstall and the 4.6 or later install needs these settings to be manually updated after the installation has completed.

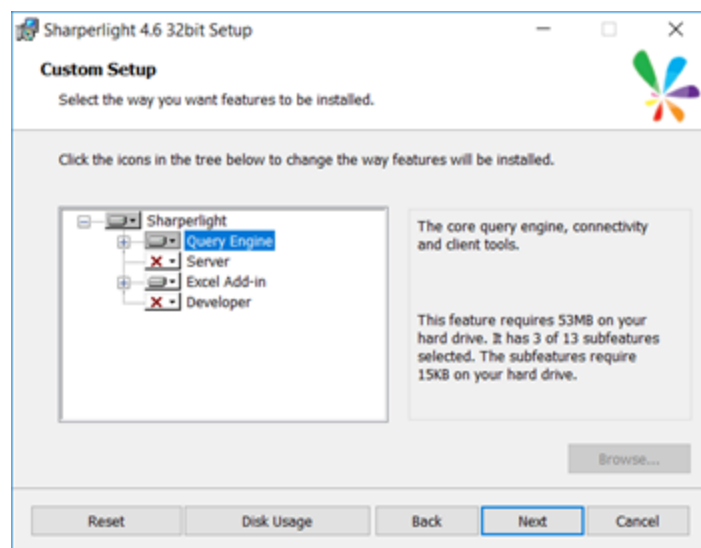
The new installer will remember the existing installation state and a subsequent release can be safely installed over the top, with the existing components upgraded to the later build.



Installer - Setup Types

The big change to the Setup Wizard is the inclusion of Setup Types, these will control what is actually installed and configured on the destination machine. A Typical setup will only install the core query engine, client tools and it includes the Sharperligh XL Add-in. The Complete setup installs everything including the datamodel development tools and it registers the Sharperligh Service in the Windows Services. To fully control the installation footprint, the Custom setup has selection control on individual features.

The Custom setup features are also available if the Sharperligh installer is run from a command line. There are a switches that can be added to the command line which will nominate the features to be installed. For more details refer to the Preparation - Command Line section in the Sharperligh 4.6 Install Guide.

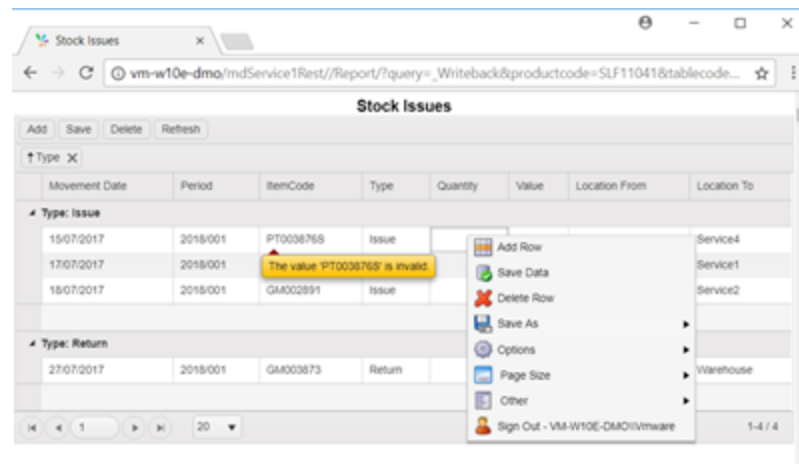


Installer - Custom Setup

2.3 HTML5 Data Entry Grid

The Sharperligh Web Service has been relying on a Silverlight control to provide grid entry for web writebacks. Silverlight has been deprecated by Microsoft and the latest web browsers are no longer supporting it. To address this loss in functionality, the HTML5 Table Grid has been enhanced with data entry functionality.

With the 4.6 release the native web grid entry interface will use the new HTML5 data entry grid. In data entry mode the HTML5 grid supports grouping and advanced filter. The new writeback grid will return error messages on the entry cell and this is an improvement because the Silverlight grid returned the messages in a status bar at the bottom.



HTML5 Data Entry Grid

3 Enhancements

Enhancements to Sharperlight are often the result of customer and partner feedback. We appreciate this input into the development cycle and we add our own ideas from direct consulting experience and carry forward refinements that could not be finished in time for the last release.

3.1 Excel Add-in

The Sharperlight XL Add-in provides direct integration into Microsoft Excel where it creates intelligent formulas that refresh queries from datamodel data sources.

3.1.1 Writeback Options

Following the release of the Foundry in October 2016 there has been a lot more interest in Bulk Writebacks in Microsoft Excel. Feedback from consultants has been very informative and four new settings have been added to the Processing Options of a writeback template.

There is now a setting to "Skip Zero values for Matrix Writeback". A writeback with input fields referenced down and across the worksheet have special matrix logic to create separate records for each intersect of row and column. The default matrix behaviour is to ignore empty cells but zero values have traditionally been included in the writeback. This new skip zero setting will treat zero values like empty cells and ignore them.

Fields	Options	Processing Options
Processing Options		
Break On First Row Error		True
Delete Based Unique Keys Only		False
Delete Mode		False
Skip Zero values for Matrix Writeback		False
File Output Options		
Output to CSV		False
CSV Filename		
CSV Delimiter		
Output to Fixed Width		False
Fixed Width Filename		
Logging		
Logging Cell		=TextBox 1
Rows Affected Cell		
Success Cell		
Validation		
Validation Cell		
Append Validation Errors to Template		false

Writeback - Processing Options

File Output to CSV has available for many years but the CSV Delimiter has been extended to include shortcodes for Tab, Comma, CR (Carriage Return), NL (New Line), CRNL and Space. The Logging Cell now allows logging messages to be written to a worksheet shape by using the unique name of the object. For example, a text box with the name TextBox 1 could be reference using an equals and the name.

The setting "Append Validation Errors to Template" is engineered around the inner-workings of a Foundry project. If the append errors to template is set to True, writebacks will copy invalid reference values to the related export table worksheets. The idea is that the invalid reference information can be fleshed out and can be loaded separately using the export table template.

3.2 Query Builder

The Query Builder is the common interface for building and refining queries. Query templates can be re-used in all the presentation layers and the same query interface is used throughout the solution.

3.2.1 Faster Expression Engine

In the move to .NET 4.6.1, code was reviewed and it provided a good opportunity to revise and update routines that had been static for several years. The expression engine is now 40% faster due to code changes.

3.2.2 Wildcard Search

The standard lookup now has a wildcard search checkbox in the Query Builder and when the lookup is opened in the web. Enabling the wildcard option will change the refresh button to only return rows that have the entered search string in any of the lookup columns. This includes searching description and reference columns. With the wildcard search it allows the additional columns in a lookup to be used to refine the selection.

Account Id:

Sales ☒ Use wildcard search

Account Id	Account Name	Account Type
<ALL>	All	
2201	Sales Tax Payable	Liability
5101	Cost of Sales from Solar Panels - 48 Cell Panels	Expenses
5102	Cost of Sales from Solar Panels - 60 Cell Panels	Expenses
5103	Cost of Sales from Solar Panels - 72 Cell Panels	Expenses
6202	Sales Salaries	Expenses

Wildcard Search Checkbox

3.2.3 Drill To

One of the more challenging feats with web reporting is adding an expression to a published query that adds a button or icon that will open another report. This is called a Drill To and there are several expression templates that show different patterns but adding the filter parameters and tweaking the web presentation can be quite fiddly.

To simplify the creation of a Drill To expression, there is now a wizard that will assist users in selecting the published query, setting the filters and adding the labels. The Drill To wizard is available on the right hand click menu in the Query Builder Output pane. The user has to select the drill to published query and it will use fuzzy logic to map filters and outputs in the parent query to the filters in the child query. The user has options to add an icon, set how the child report will be displayed, add titles and a tooltip to the icon in the parent report.

Drill To

Attributes

Report Code: AAA.DrillToReportChild

Custom Title:

Content Tooltips:

URL Parameters:

Prompts:

Type: ☒ Icon ☐ Button ☐ Hyperlink

Display Content: ☒ In new Tab ☐ In same Tab ☐ In overlay Window

Content Title:

Filter Name	Description	Map to
@Comid	Company Id	@Comid - Company Id
@PerNo	Period No	@PerNo - Period No
@AccId	Account Id	@AccId - Account Id

Link Filters: ☐ Filter ☐ Output

Map to:

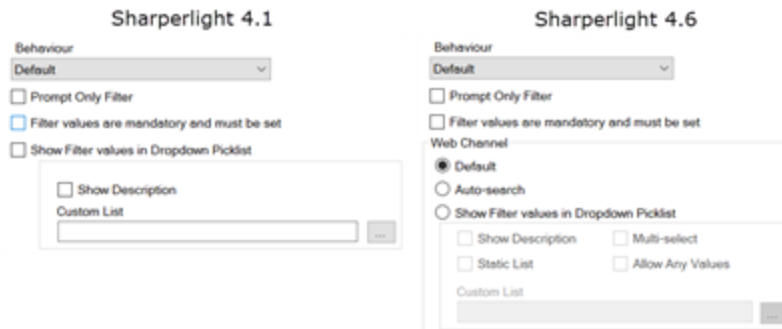
Drill To

3.3 Web Channel

The Web Channel is an overall term to describe the web content rendered by the Sharperlight Web Service. It is primarily a viewing and filtering interface to access the published queries through a web browser.

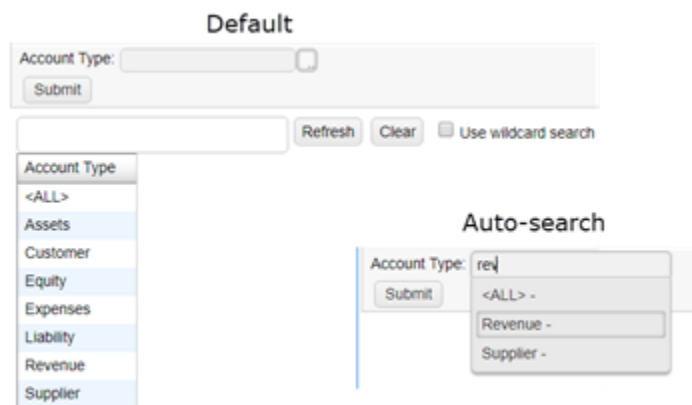
3.3.1 Auto-search and Multi-select

The standard filtering in the Web Channel includes a date picker, lookup lists displayed below the prompt bar and dropdown picklists that emerge below the filter input box. With this release the filtering in the Web Channel has been extended to include options that allow auto-search in the filter input box and multi-select filtering. To allow these options the bottom left hand corner of the Filter Options Attribute tab had to be re-organised and extended with radio buttons and check boxes.



Filter Options

In the Web Channel a published query with a filter set to default will use the fields data type and datamodel settings to generate the appropriate filter interface. For example, a field containing codes which has a description field defined in the datamodel will display a standard filter with a lookup button on the far right of the input box. The new Auto-Search option removes the lookup button, the user has to type the first few letters of the required value into the filter input box and a dropdown will appear, with the lookup values filtered using the letters as the starting point. To select a value in the dropdown use the down arrow to navigate to the required value and then hit Enter to accept the highlighted value.



Default and Auto-search

To use a Multi-select filter in the Web Channel, change the filter to use the Dropdown option and then enable the Multi-select checkbox. In the web the input box for the filter will double in size and it will behave like a dropbox but each time a value is selected it will be added to the filter. Each filter value will be suffixed with an X and this icon can be used to delete individual filter values.

Multi-select

3.3.2 Browser Resource Caching

The Sharperlight web interface uses hints in its web content to encourage web browsers to locally cache resource files. Re-using previously fetched resources can greatly improve retrieval time and minimise the download bandwidth. With this release the hints have been revamped and extended to improve the caching with Microsoft Edge and Google Chrome.

3.4 Foundry

The Foundry is a rapid application development environment for defining data structures and business rules, the interface can then transform the project in a datamodel and auto-generate a web interface for data maintenance.

3.4.1 Auditing

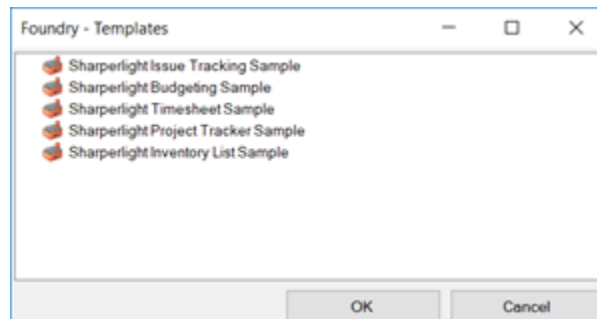
An Audit checkbox has been added to the Tables tab in the Foundry. When enabled the related table will have a duplicate table created in the destination database with an `_Audit` appended to its name and it will have additional columns for audit date, audit change type and audit row id. With auditing, changes in the primary table will be copied into the audit table with the date and time of the change, along with the unique row id of the original record. The audit table becomes a query table in the datamodel and it will be automatically added an Audit tab on the Foundry projects main menu.

Audit Date	Type	Description	Audit Type	Audit - Row ID
2017/06/16 15:45:53	Issue	Stock Issue	Insert	1
2017/06/16 15:46:07	Return	Stock Return	Insert	2
2017/06/16 15:46:48	Default	Default	Insert	3
2017/06/16 15:46:59	Default	Default	Delete	3

Audit Menu

3.4.2 Templates

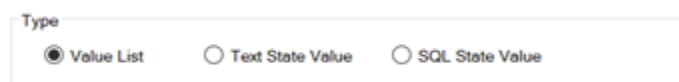
When Foundry was released in Sharperlight 4.0, a starter pack of Foundry projects were supplied to partners, to give them them an idea of what it could achieve. These samples proved very popular and the projects for Issue Tracking, Budgeting, Timesheeting, Project Tracking and Inventory; can now be open directly in the Foundry interface using the Templates button.



Foundry - Templates

3.4.3 State Values

Text and SQL State Values have now been added to the Foundry. They share the same tab as the Value Lists and the tab title has been changed to Values, to recognise that it now covers more values.



Foundry - Values

3.5 Studio

The Studio is the advanced design interface for developing and maintaining datamodels.

3.5.1 FoxPro Connection Template

Microsoft Visual FoxPro has been out of development since October 2007 but there are still a few applications that use the FoxPro database engine to store back-end reference and transactional tables. With the level of interest in FoxPro, this release includes a connection template to the FoxPro ODBC driver. The only caveat on using this connection template is that it will only work with a 32 bit build of Sharperlight because the FoxPro ODBC driver is only available in 32 bit.

3.5.2 ODBC DSN Connection Templates

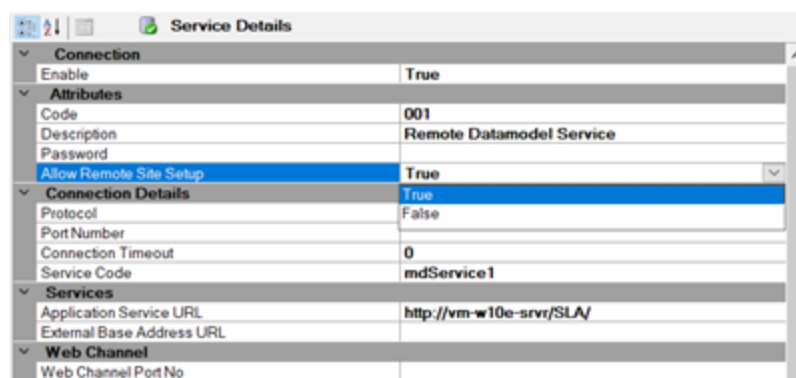
To cover the multitude of ODBC drivers and connection types, this release includes System and File ODBC DSN connection templates. These use a generic connection string and attempts to use ANSI-92 SQL syntax. If the underlying database engine uses non-standard functions or SQL syntax, the connection settings can be modified in the Studio.

3.6 Site Setup

Site Setup stores the licensing, user details and security settings that control the delivery of Sharperligh.

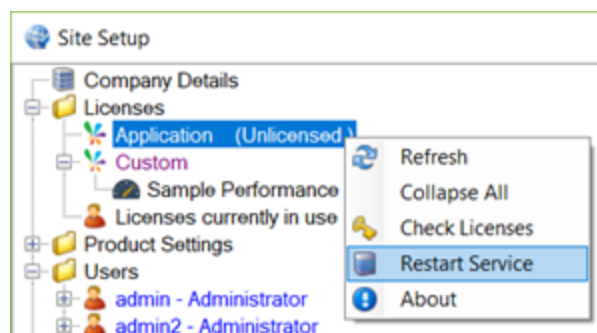
3.6.1 Remote Access

In previous releases, access to Site Setup was limited to client installations that had local database connections to the Sharperligh System database. With this release, there is now an Allow Remote Site Setup option in the service settings in Client Setup. This means an application server hosting the Sharperligh Service can be configured to allow remote connections to Site Setup. The Allow Remote setting in Client Setup defaults to False and this maintains the legacy behaviour of blocking remote connections to Site Setup but it can be toggled to True and a service restart will make remote access available.



Allow Remote Site Setup

Opening Site Setup on a remote client connection, still requires admin rights to Sharperligh. The functionality in Site Setup on a remote connection does not include some of the more advanced features like assigning a group to a selection of users. However, it does have the option to restart the service to take effect of configuration changes.



Restart Service

3.7 Other

3.7.1 Display Scaling

With the popularity of high resolution monitors, there has been feedback from customers, that the Sharperligh desktop user interface was blurry, miss-sized and the text was too small to read on 4K and UHD displays. With this release, significant time has been spent testing and tweaking the UI components to optimise them for high DPI resolutions beyond 1980 x 1080 (1080p). After trialling

the changes with different 4K monitors and TV's, the Sharperligh desktop interface can now adapt to ultra high definition displays like 3840 x 2160, with a scaling of 125%, 150% and 200%.

3.7.2 File Associations

Most Sharperligh desktop tools have a menu option to save individual templates or to export a packaged selection of templates. The templates might be published queries, dashboard pages or scheduled tasks but the individual templates can be loaded back into the application and the packaged templates can be imported. As part of the new installer wizard, Sharperligh file types have been associated with there source applications and double clicking on these files will display the related open and load interface. The user must still have a valid login and they will be challenged with a message window to confirm that they want to load the contents of the file.

File Extension	Description
qry	Query Builder query
pbq	Publisher published query
pbqlist	Publisher packaged list of published queries
dbp	Dashboard page
dbplist	Dashboard packaged list of pages
task	Scheduler task
tasklist	Scheduler packaged list of tasks
mqt	Materialised Query table
mqlist	Materialised Query packaged list of tables
rpr	Report Rows row definition
rprlist	Report Rows packaged list of row definitions
ctb	Custom Table definition
ctl	Custom Table packaged list of definitions
lic	Site Setup license file
meta	Studio datamodel
fdy	Foundry project

3.7.3 Backup

The Application menu now has a Backup button that will create a compressed archive containing an export of all the main configuration tables and settings in Sharperligh. The back files are created with the naming convention SL_BAK_YYYYMMDD_HHMMSS with a ZIP file extension. The backup includes published queries from Publisher, Dashboard Pages, Scheduled Tasks, Report Rows, Materialised Query databases, Custom Tables and Custom Fields.