

2017-08-14 Version 9.2.3

1. The “shadow” drop-down arrow on *ChooseFromList* drop-downs has been vanquished.
2. The sequence of execution of *OnExitSet* with *ChooseFromData* has been improved.

2017-08-10 Version 9.2.2New Features

1. The *Rule* command can now be applied to all repeaters, not just those captured in a grid.
2. Execution of *LinkToOtherData* changed from last of all the exit events to just before *LinkToDataSet* and (if no *LinkToDataSet*) before *OnExitSet*.

Bug fixes

3. When the *CaptureFromDataElements* control was in a Tab, the drop-down was not being populated.
4. The result of functions *Chr(10)* and *Chr(13)* in the text in *InsertFormattedText* will now function identically.
5. When a *ChooseFromRDBList* appeared immediately after a *ChooseFromDataSource* then combined with *Use Other Data* the behaviour was not correct.
6. The evaluation of non-delayed *Rules* on the *ChooseFromDataSource* was being done too early - before the data had been retrieved.
7. The application of *Refresh* on the data source commands was not being done - it was only being done for the *RefreshSave* option.
8. The first control in the desktop interview was not getting focus.
9. There were problems with Preview functionality which have been corrected.

2017-07-05 Version 9.2.1

1. When interpreting configured saved file names, dots are no longer replaced with underscores.

2017-07-04 Version 9.2.0

1. There is a Wild Card feature in the search for templates by content.
2. The *GetDataSourceData* and *GetXMLElementValue* functions can be used to retrieve values from data sources during the assembly of the document. This is different to the existing data source commands, all of which are required to acquire the data before the assembly process commences.
3. *SetDataSourceData* can be used to set data field values in a data source.
4. *SalesForce* date handling was improved to take into account time zone considerations.
5. The handling of chained *Rule* commands has been improved so that when a Rule in the chain is broken, only the message relevant to that Rule is shown.
6. The Desktop implementation of *Rule* will do the application of the Rule command as focus leaves a control on the interview, rather than (or, actually, in addition to) after the OK button has been clicked. In order to ensure backward compatibility with existing Rules which need to be executed once all the data have been captured, the “delayed” option has been added to the “hardness” argument: e.g.
«*Rule(ParentName,hard/delayed,(count(Child) > 0),‘Please make sure there is at least one Child.’)*»
7. The *DatePicker* control on the Desktop will display the calendar in the language chosen by the user in the Ribbon. It will also display the date in the control in the “short date” format as determined by the language chosen in the Ribbon. For example, if English US is chosen, then the date is shown as, for example, “5/25/2017” whereas all other languages will show the date in the format “25/05/2017”, with the preferred date delimiter for the major region where that language is spoken.

8. When too many arguments are passed to a *Define* command, a nicely formatted message is given.
9. If an empty (rather than *null*) value for the data source *Id* is provided, then instead of an error message being given, an empty XML data set for that data source will be provided.
10. The key combination <Ctrl + Alt> in the Desktop interview Help section will no longer provide context. Rather, use <Ctrl + Enter>
11. A Stored Procedure (SQL Server) can be used as a Data Source.
12. The configured empty marker was not being coloured correctly. It now is.
13. A problem with Highlight Inserted text was that if the text surrounding the fillpoint was already in some non-“Automatic” colour, then the Highlight did not work correctly.
14. When a fillpoint occurs in the dataset, XpressDox will try to fill it. If there is a syntax error in the fillpoint, an error message is displayed, and the document will not be assembled. This is as it has always been. Now a change has been introduced so that the document will be assembled, but the content of the data element which contained the failing fillpoint will be filled with an error message, hopefully helping with trouble shooting.

2017-04-19 Version 9.0.3

There was a problem with «[CaptureDataElement\(XXX, Decimal\)](#)» in that if there was one or more spaces before the last argument “Decimal” (or “Text”), then that last argument was not recognised as a data type, but was treated as a data element name. This is now fixed.

2017-04-12 Version 9.0.2

This version contains mainly bug fixes, with the following useful new feature:

1. Especially with regard to the Clauses features, it is now possible to define shortcuts to common folders from other folders. You can find the documentation for this feature at [Define Folder Shortcuts](#).

Bug fixes

2. Windows 10 makes some dropdown boxes look read-only. The style of the dropdown boxes has been changed to subvert that new Windows behaviour.
3. A set of circumstances involving data sources and grids resulted in some grids acquiring repeated empty rows. This has been fixed.
4. If a fillpoint was highlighted using the Word highlight feature, then the highlight was not maintained when the template was “painted” by XpressDox. The painter now preserves highlighting.
5. The *NumberPhrase* function and functions that use its functionality (*Dollars*, *Pounds*, *Rand*, etc.) do not support negative values. Instead of issuing a Windows message saying words to the effect of “Invalid Number Format”, the function will put a message into the assembled document highlighting the value in question. This will make it easier for the template author to take that situation into account.
6. Sometimes the Windows/Word configuration settings file that XpressDox uses becomes corrupted. There was some code in XpressDox that handled that situation, but not enough. That functionality has been extended.
7. The *Save Data Only* checkbox in the Desktop interview was becoming invisible when the *Save Assembled Document* checkbox was invisible. The *Save Data Only* option is now always available.
8. An & in a Radio Button or Check Box label would be removed. This is no longer the case.

2017-03-13 Version 9.0.1

1. The bug in the *When* command which rendered an apostrophe in the “false” part as ' has been fixed.

2. The command [UpdateTableFormattingForPdf](#) has been added to assist with Word table formatting issues when the document is saved as PDF.
3. The function [GetVOrDefault](#) will appear to enable retrieval of the value of a variable even before the variable has been assigned a value.
4. A few rather esoteric bugs have been fixed.

2017-02-28 Version 9.0.0

1. The biggest new feature in this version is actually a group of features all revolving around the format of numbers in the XML dataset. It is now possible for the template author to tell XpressDox to store all numeric values in XSLT format. The reason for this is that the template author will now know that numeric data elements can be used in calculations without having to use the RegionToX function (and then XToRegion to get the result back to regional format for use in other functions). It will also be possible for the template author to use XSLT functions like sum, floor and ceil. The new commands are listed below, but a full description will be found on the web site:
 - a. The «[ExpectXSLTFormatNumbers\(\)](#)» command.
 - b. A re-implementation of the «[CaptureDataElement\(...,Decimal\)](#)» feature.
 - c. A new option that a user can select in the Ribbon before running a template, called *Interview Number Format*, which the user can use to select which format s/he would like to use when entering numerics in the interview.
 - d. The command «[ForbidThousandsSeparators\(Yes\)](#)» which will not allow the user to capture numbers which contain the thousands separator. For example, the number 1234.56 will be valid as input, but not 1,234.56 (for US/UK English format settings).
2. A major new trouble shooting feature has been added for template authors. The «[Log\(\)](#)», «[LogDataElements\(\)](#)» and «[LogVariables\(\)](#)» functions will cause trouble-shooting information to be written to a log file during template execution. This log file is called *LogXpressDox.txt* and will be written to the *My Documents* folder of the user running the template.
3. Many of the XpressDox forms did not display correctly on high density monitors which are being shipped more frequently with laptops and other configurations. This has been fixed.
4. In the desktop interview, the date calendar will display in the language of the interview, rather than, as was originally the case, in the Regional Settings language.
5. The feature in the XpressDox Explorer which searches for templates with specific content has an option to do a case-sensitive search, and also an option to include or exclude the *XDTemplateVersions* sub-folders which are created when the configuration has been set to apply template versioning.
6. The command «[InsertHyperlink\(\)](#)» will insert a hyperlink into the document.
7. The preview of the partially-assembled document in the Interview is available to all users, not just to template authors. The document is no longer opened in Word, and so, for instance, cannot be saved in that incomplete state.
8. There is a new function for handling variables. «[GetVOrDefault\('X',1\)](#)» will get the value of variable 'X' if it has been set, but if variable 'X' has not yet been assigned a value, then it will be assigned the value 1 (in other words the second argument to the function, which is 1 in this example) and that value returned into the document.

Bug Fixes:

9. A bug in the *Debug* command implementation which caused the template running to fail has been fixed.
10. The error message about an *Else()* needing to be on its own paragraph had an extra closing parenthesis in it, i.e. *Else())*. This has been fixed.
11. There was a bug in the *Required* command that if there were exactly two data elements specified in the command, the second one would be ignored.

12. *AlignCaption* of Top was behind two glitches in the Desktop interview, which have been fixed:
 - a. In a *Horizontal Grid* which had a *Heading*, the top-aligned captions were mis-aligned.
 - b. In some instances the interview would not be displayed properly - the captions were showing but the controls underneath them were invisible until the user re-sized the form.
13. A number of intermittent errors resulting in the user being asked to send a troubleshooting email to XpressDox Support have been fixed.
14. Problem with *DropDownPanel* which had *Rules* on data elements has been fixed.
15. A problem when using the “&” character in the true or false value in the *When* command has been fixed.

2016-11-18 Version 8.4.0

1. A bug in the *RunTemplates* command giving rise to a Windows error “Index out of range” has been fixed.
2. “Windows” has been added as a language choice in the ribbon. It will mean that the Regional Settings as set in the Control Panel will be used for the language (where the language is supported by XpressDox) as well as number formats. This will be useful for users who are not in English-speaking areas.

2016-10-14 Version 8.3.3

1. A bug in *LinkToDataSource* giving rise to a “null reference exception” has been fixed.
2. The Rich Text Editor has been delivered correctly with the install of XpressDox.
3. The Preview area in the Explorer was not showing correctly. This is now fixed.
4. The *RaiseToPower* function is introduced.

2016-10-05 Version 8.3.2

1. The *Decimal* option in «[CaptureDataElement\(Price,Decimal\)](#)» will automatically insert a *Rule* into the interview to allow only numeric values to be captured.
2. When a template name is used in an *IncludeTemplate* command, then normally the extension *.xdtpx* is not required in the file name. However, if the file name itself contains a dot (‘.’) then the extension must be provided. The Command Editor wizards now contain that information in their *Help* text.
3. A bug around executing two or more *MergePDFForm* functions without having to re-load Word is now fixed.
4. The user interface for non-authors now includes German as a supported language (along with English, French and Afrikaans).
5. The interview component (in the Desktop version of XpressDox that is used for the *CaptureAsLongText* command will now support nearly all the editing functions of Word.

2016-09-12 Version 8.3.1

1. New underlying code has been introduced to improve assembly of very large documents.
2. The “Delay” option has been added to the *RunWordMacro* command.

2016-09-07 Version 8.3.0

1. The command «[WriteTimeStampToDataSet\(Yes\)](#)» will result in a time stamp data element being written to the dataset after the assembly of the document, and this will contain the start and end times of the assembly and the number of milliseconds taken.
2. A bug in the *SetSaved...* commands to assembled files, folders and data sets has been fixed. (When specifying XSLT or XpressDox functions as the argument to the command, the function was not being evaluated).

3. There is a function similar to *PrefixWith* called *SuffixWith* that appends a string if the source string is not empty.
4. Some issues were introduced in version 8.2.1 which caused numerous data elements used in the saving of documents and data sets to be left in the dataset. These elements are now cleared out before the dataset is saved.

2016-08-26 Version 8.2.1.27

1. There was a problem with the license installation form in version 8.2.1 . This is now fixed.

2016-08-24 Version 8.2.1

1. There is now a “Save as PDF” button in the Utilities group in the XpressDox ribbon. It will save the current document as an encrypted (read- and print-only) PDF file.
2. Start and end time of assembly will be written into the XML dataset.
3. A bug causing Word tabs to, in certain circumstances, surround text inserted by fillpoints, has been fixed.
4. Empty tables (i.e. tables with no rows) could be created by conditional logic, and would sometimes cause the assembled document to be corrupt. Empty tables are now deleted from the assembled document.
5. When the Regional Settings has set a character other than “/” as the separator in dates, then «[FormatDate\(Date, 'dd/MM/yyyy'\)](#)» would display that other character instead of “/”. This might be what is desired, but, if not, then using “//” in the date format string will force the “/” character inside the rendered date: «[FormatDate\(Date, 'dd//MM//yyyy'\)](#)»
6. Wherever the *<DataElementName>* syntax is used in templates or in the configuration, you can now have expressions between the < and >. For example, to create a folder with today's date in it, you could have something like “..\DataSets\<Today('MMdd')>” in the configuration, rather than having to create a data element with a *CreateDataElement* and refer to that in the configuration.
7. A bug in parsing of *BaseTemplate* with *<DataElementName>* syntax has been fixed.
8. An attempt (hopefully successful) has been made to inhibit “Call rejected by callee” errors.
9. Some bugs relating to running of macros have been fixed.
10. The new system Rule function *IsUKPostalCode* has been added.
11. Where necessary, the Command Editor wizards have been modified to include a check for valid data element names to trim down on errors in commands.
12. The XpressDox Save as Template button in the ribbon would save a template even if the user pressed the “Cancel” button. This has been fixed.
13. The Edit Fillpoint function in the ribbon would not work correctly if the fillpoint had an invalid data element name in it. This has been fixed.
14. The function «[LastDayInMonth\(Date\)](#)» will return the last day of the month in which *Date* falls.

2016-07-01 Version 8.2.0

1. The XpressDox Explorer folder browser works similarly to the Windows file system explorer introduced with Windows 7.
2. The command «[HideInsideIV\(\)](#)» applied to a data element will make sure that the data element is not visible in an interview, but will nonetheless be able to take part in features such as condition evaluation which otherwise require the data element to be present in the interview.
3. The data source implementation has special handling for null values in database columns which are used as foreign keys in complex collections.

2016-06-13 Version 8.1.5

1. A bug when using a “range” in a data source where the value of a column in the range includes a single quote has been fixed.

2016-06-10 Version 8.1.4

1. Loading of the Preview in the Explorer has been improved.
2. There was a bug which caused the creation of a data sheet to fail. It has been fixed.

2016-06-06 Version 8.1.3

1. XpressDox now uses .NET 4.5.2. The installer will attempt to use the Microsoft install facility to make sure that your system is up to date with this version of .NET and also the Visual Studio Tools for Office Runtime. (Please note that Windows XP does not support .NET 4.5.2, and so this version of XpressDox, and all later versions, will not run on Windows XP. Support for Windows XP was discontinued by Microsoft in 2014, and using Windows XP is now considered dangerous from a virus protection and security point of view).
2. There is a Preview feature in the XpressDox Explorer. Which is particularly nice, seeing it recognises XpressDox templates, which the preview in the Windows Explorer does not (although it could).
3. A slightly modified licensing paradigm means that all licensed users will be able to download and install the latest version. The new features introduced in each version will only be available to users with a license which supports the latest version, but all bug fixes, and of course all the features related to their license version, will be available.
4. The «IncludeTemplate/Picture/Spreadsheet» commands now support the <DataElement> syntax. For example: «[IncludeTemplate\(LetterHead<Department>\)](#)», which would work if the *Department* data element were known before the interview is constructed (e.g. if it were retrieved via an «[IncludeDataSourceData\(\)](#)» or similar command). As in the past, if *Department* is only known after the interview, then «[InsertTemplate\(concat\('LetterHead', Department\)\)](#)» would be needed.
5. XpressDox will now create a “data sheet” for each template saved. This is controlled via the configuration. On the first tab of the configuration (Standard Folders) is a new area at the bottom enabling you to indicate that a data sheet should be created. Then the command «[MergeDataSheetTemplate\(\)](#)» in a template will cause its data sheet to be merged after the template itself (as a separate document). Further options are available, for which see the help on the web site.
6. The Search facility in the XpressDox Explorer can be selected inside a folder, and will also list the files found, rather than opening them.

2016-04-29 Version 8.1.0

1. A bug in *RemoveColumnIf()* has been fixed.
2. A bug in *RunAsHugeTemplate()* has been fixed.

2016-04-26 Version 8.0.2

1. Problems with the location of the license file have been addressed.
2. There was a bug introduced with version 7.3 with the default item in the *ChooseFromList* combo-box. This has been fixed.
3. The *FileExists* function has been introduced.
4. Some issues with non-English languages have been fixed.
5. The *RemoveColumnIf* command has been enhanced so that you can provide a column whose width must expand when this column is removed, or else the column can be emptied instead of deleted.
6. The *PrefixWith* function will output nothing if the value passed is empty, otherwise it will output the value prefixed with a specified prefix.

2016-04-05 Version 8.0.1

1. The XpressDox interview is no longer “modal” - which means that when an interview is active the user can operate inside Word. New documents can be opened, even the template being run can be opened while the interview is still visible.

Because of this it is now possible to preview the assembled document before the interview is completed. There is a button on the interview toolbar which will be visible. This is only for template authors.

2. The XpressDox Explorer has an option to search for templates containing one or more strings. This is only available to template authors, and appears when <Right Click> is pressed on a folder in the Explorer when it is selected to open a template. The option is called “Search for Template with Content”, and presents a user interface to guide in the choices available. Templates found in this way will (optionally) be opened in Word.
3. The command «[PaintInsertedText\(Yes,Red\)](#)» will cause all text which is inserted via fillpoints in the assembled document to be coloured Red. The Command Editor wizard will present a mechanism for choosing the available colours so that you get the name right. Then, in the Common Tools menu in the ribbon, is a button called Find Inserted Text, which will find each piece of inserted text throughout the document.

You can also allow the user to choose whether they want this painting or not by something like:

«[ChooseUsingCheckBox\(PaintInserted,Y,N,N\)](#)»

«[PaintInsertedText\(Yes,Red,PaintInserted = 'Y'\)](#)»

This functionality replaces the original feature set described in point number 13 in the features for version 7.3.0.

4. The relative-path resolution for Document and Data Save paths and Helper folders can be set to indicate “relative to the folder being configured” rather than the folder from which the template is being run. This would typically be done at a high level in the folder tree so that it affects all those folders lower down in the tree. The setting for this is in the More Settings tab in the folder configuration form.
5. There is a setting in the More Settings tab in the folder configuration form which will enable template versioning. Whenever a template is saved using the XpressDox Explorer, then the earlier version (if there is one) will be copied to a sub-folder called XDTemplateVersions. If there are already versions of that template in the XDTemplateVersions folder then any new versions are numbered 0001, 0002, etc.

Standard Folders | Helper Folders | Standard Data Elements | Data Sources | Other Settings | **More Settings**

Folders Relative to This Folder

This setting refers to the way XpressDox resolves relative file paths.

An "absolute" file path is one like "C:\XpressDox Templates\SubTemplates\Letterhead.xdtpx", whereas a "relative" file path would look something like ".\SubTemplates\Letterhead.xdtpx".

The general which XpressDox applies when resolving relative paths is that the paths are regarded as being relative to the folder in which the template being run is found. In the above example, the relative path .\SubTemplates\Letterhead.xdtpx would resolve to "C:\XpressDox Templates\SubTemplates\Letterhead.xdtpx" if the running template were in C:\XpressDox Templates.

When this rule is applied to folders defined in the configuration (such as in the Standard Folders and Helper Folders tabs), this setting here can be used to say that the "base" folder relative to which the absolute should be worked out is not the template's folder, but the folder being configured here.

Make Folders Relative To This Configuration Folder

Template versioning

If this option is selected, then a folder called XDTemplateVersions will be created as a sub-folder of the template folder, and all old versions of templates will be stored there whenever a template is saved.

Apply Template Versioning

6. «[RemoveRowIf\(<condition>\)](#)» and «[RemoveColumnIf\(<condition>\)](#)» can be placed inside a Word table cell, and the relevant row or column will be removed if the condition is satisfied.
7. XpressDox can now record every usage of a template, giving the date and time it was run, along with the user who ran it. This requires a database (probably a formal one like SQL Server or MySQL, not Excel or Access) with a table in it which has the following columns:
 - An Id - 36 characters long as XpressDox will set this using a GUID.
 - LogEndTimeStamp: a date/time value
 - LogStartTimeStamp: ditto
 - LogTemplate: the full file path of the template, so should be at least 256 characters long.
 - LogUsername: the user name with which the user logged on to Windows.

Then a data source must be defined for this table, typically in the global “Office settings” configuration, or at least at a folder including those folders with templates whose usage needs to be recorded. The data source should have the name XDUsageLog (it is a choice in the drop-down for the data source name in the configuration form). The features “Use for All Templates” and “Allow Insert” must be chosen. XpressDox will choose these automatically when the data source name is XDUsageLog and when you tab out of the Name field in the configuration form.

That is all that is required - after the data source is configured, and probably after reloading Word, XpressDox will proceed to record the template usage.

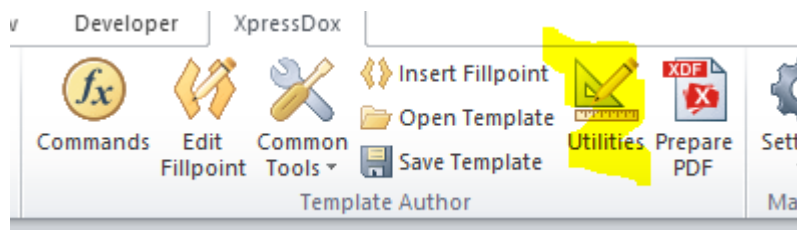
8. The Command Editor will now dock inside the Word document area rather than floating outside Word.
9. It is possible to colour the inner text of an Empty Marker. The Command Editor wizard will help. For example, «[SetEmptyMarker\(\[-cyan-**-\]\)](#)» will result in an empty marker showing in the assembled document as [**].
10. The «[ShowHidden\(\)](#)» command overrides what the user chooses with the “Show Hidden Fields” on the interview. «[ShowHidden\(Name,Address\)](#)» will cause the Name and Address fields to be shown on the interview, even if the relevance rules would otherwise cause them to be hidden. «[ShowHidden\(\)](#)» with no parameters has the same effect as the user choosing the “Show Hidden Fields” option on the interview.
11. The “CaptureDataElement” part of the *Define* command now has the default value of “None” enabling setting of the *ReadOnly* and *Required* sub-options without forcing the capture of the data element.
12. With «[PositionOfRepeater\(\)](#)» it is possible to test the position of the current repeater within interview commands such as *OnExitSet*, *ReadOnly*, etc.
13. If the first control on a repeater is a *ChooseFromList* and the interview has some non-empty data (e.g. from the user pressing Use Other Data) and the value of the control for the first repeater item is the default (first value in the list of items to be chosen), then the caption in the treeview was not being constructed properly. This is fixed.
14. There was an issue with regional settings and *Include/InsertPicture* which has been fixed.
15. *IncludeFileData* with the *NoRefresh* option will not include the file data when Use Other Data is selected in the interview.
16. When custom document properties from the “source” template are merged into a “base” template (i.e. the template referred to in a «[BaseTemplate\(\)](#)» command), then if the custom document property appears in both templates, the property in the source template will be copied into the assembled document with “Source” prefixed to the property name..

2015-10-06 Version 7.3.1

1. Enable numeric only caption on Tab.
2. Accommodate Office 2016 in install.

2015-09-18 Version 7.3.0

1. A document assembled by XpressDox can now be saved as HTML with the command «[SaveAsHTML\(Yes\)](#)».
2. The *SaveAsPDF* functionality will now save all PDF documents in encrypted form so that they can only be viewed or printed. If it is required to save PDF results in un-encrypted form, then the new *Unencrypt* parameter must be provided to the command, e.g. «[SaveAsPDF\(Only,Unencrypt\)](#)».
3. *OnEnterSet*, *OnExitSet*, *LinkToDataSourceOnEnter* and *LinkToDataSource* events are executed in the sequence in which they appear in the template (except that all the events associated with a specific command are executed one after the other).
4. The «[Format\(\)](#)» function is introduced. It will make concatenation of a mixture of fixed text and dataelement or function values easier to visualise than using a «[concat\(\)](#)» function would do. For example, a «[CreateDataElement\('File',concat\('Acc - ','AccountNumber,' - ','Today\('yyyyMMdd'\)\)\)](#)» would look like this:
«[CreateDataElement\('File',Format\('Acc - {0} - {1}','AccountNumber,Today\('yyyyMMdd'\)\)\)](#)»
5. A bug in the *Required* function when there are two data elements supplied to the command has been fixed.
6. You can inhibit the automatic saving of the data set (XML) file with the command «[SaveDataset\(No\)](#)».
7. Support for various new features requires that every template which is run after this version has been installed will have the artefacts re-created. This will be unnoticed unless at your site users do not generally have Windows file permissions to write to the XDartifacts folders. However, a template author (who typically should have those permissions) using the “Check Syntax of Templates in the Folder” option in the XpressDox Explorer can perform this operation (i.e. writing to the XDartifacts folder) for all the templates in the selected folder.
8. A bug in the functioning of the *ChooseFromFile* command has been fixed.
9. The *Required* option did not work previously for repeating data elements. It now does work.
10. The condition supplied in the *ReadOnly* command is now applied dynamically and not just when the interview is constructed.
11. There is a new data source type - i.e. Web Service.
12. The function *ToHtml* will wrap a data element value (or any other string value) in the HTML tags `<html> ... </html>` so that the data value can be formatted using *InsertFormattedText*.
13. There is a feature which will highlight the text inserted by a template. This will help with trouble shooting a template. The template can be “decorated” to do this highlighting, and then before releasing the template for general use, the highlighting can be undone. The default highlight colour is blue, but this can be configured. This is all found in the *Template Painting* tab in the *Template Author Utilities* which is accessed via the XpressDox ribbon.



2015-06-30 Version 7.1.1

1. Handling of Refresh/NoRefresh options on the *IncludeDataSource* command has been made more intuitive. The option SaveNoRefresh has been introduced for all the data source commands.
2. If a data element is changed via *OnEnter/ExitSet* and that data element belongs to a data source then the data source will be regarded as “changed” even if the element(s) are not in controls on the interview, e.g. even if they have been hidden with *ExcludeFromIV*.
3. The function *InsertFormattedText* will test the value to be inserted for HTML, and if it is in fact HTML, then the HTML is rendered into the Word document with all the HTML-specified formatting applied. The *CaptureAsLongText* command has been extended to enable the capture of data into HTML. This is achieved using the “Rich” option, for example: «*CaptureAsLongText(Address,3,Always,Rich)*».
4. A bug in *RunAsHugeTemplate* when the template contained a data -source access command has been fixed.

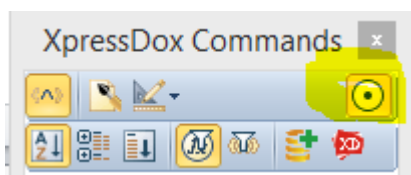
2015-05-11 Version 7.1.0

1. The list of Basic commands in the Command Editor has been expanded.
2. There are two new functions:
 - a. *IsTrue()*, which returns a true value if it is passed one of “true”, “yes”, “y” or “1” (case independent);
 - b. *TemplateName()*, which returns the file name of the currently running template.
3. It is now possible to put a button on the interview, and then attach events such as *OnExitSet* and *LinkToDataSource*. Those events are then fired when the user clicks the button. It provides a mechanism for performing those events only with explicit action by the user, rather than implicitly when a control on the interview receives, or loses, focus. The command for this is «*Button(DummyElementName)*».
4. The XpressDox Explorer has a new <right-click> feature which selects all the entries in the file list after the one on which the <right-click> occurs.
5. Some very handy but not easy to explain features were added to the “Convert from foreign system to XpressDox” utility.
6. In the configuration it is possible to specify that a file path or file name should be constructed using the contents of data elements. This is a long standing feature of XpressDox. In the past, if a data element in the configuration did not exist, XpressDox would issue an error message and refuse to save the relevant file. This no longer happens, what will happen is that an empty string will be used to replace the value of the missing data element.
7. Up to now it was necessary for the template author to ensure that any data elements used in the configuration to construct parts of a file path had values which did not contain any invalid file-name characters. XpressDox will now do this automatically.

2015-04-14 Version 7.0.0

1. The main new feature in this version is the ability to embed spreadsheets into the Word document being assembled. See the «*IncludeSpreadsheet()*» command.
2. Considerable enhancements have been effected in the automatic conversion from GhostFill.
3. New features regarding XpressDox Arrays have been introduced, such as *ArrayDelete*, *ArrayClear*, *ArraySortByIndex* and *ArraySortByValue*.
4. The *ChooseFromFile* command now selects the first entry in the file by default.
5. *ChooseFromRepeatingData* puts an empty item at the top of the drop-down so that the user can clear the selection if they choose something by mistake.
6. Some more advanced XML/XPATH features are now catered for, such as «*ForEach(Child/*)*» which iterates through all the sub-elements of the Child element regardless of name.

7. There is a new toolbar button on the Command Editor which enables the template author to toggle the list of commands shown between a Simple list and the full Advanced list.



2015-01-06 Version 6.4.2

1. Some new Array support functions.
2. There was a bug in the support for MySQL databases and the *AllowInsert* data source option.
3. The install now delivers the correct Microsoft Program Interop Assembly files.

2014-11-27 Version 6.4.1

1. «UpdateFields(Yes)» will cause XpressDox to update all the Word fields and Table of Contents in the document.
2. The updating of fields and table of contents will happen automatically when the Cloud version of XpressDox assembles a PDF document.
3. The *IIf* function has been introduced. It works in a way analogous to the Visual Basic *IIf* function, and provides a kind of short-hand around writing «If()»... «Else()»... «End()»

2014-10-02 Version 6.3.0

1. The functions *SubstringBeforeLast* and *SubstringAfterLast* will get the part of a string before a particular substring, or after it. For example, you could extract the extension of a file name with «*SubstringAfterLast*(Filename, '.')»
2. The *GetRegistryValue* function will enable extraction of values from the Windows registry.
3. A fair number of bugs have been fixed (including one where a MySQL data source would return empty sets of records)

2014-06-25 Version 6.1.2

A fairly obscure bug in the *HzGroup* usage has been fixed, and also some minor bugs (not even worth calling them bugs, really, more to do with cosmetics) in the “Edit the nearest fillpoint” functionality.

A problem with installing on Word 2007 is also addressed.

2014-05-15 Version 6.1.1

A bug in the Convert Fillpoint Delimiters feature has been fixed.

2014-05-09 Version 6.1.0

The functioning of the Empty Marker feature has been improved. In particular:

- The *SetEmptyMarker* can be placed in any part of the template, not necessarily before any data elements that it affects.
- The command *ShowFormattedMarkerWhenEmpty* needs to be used far less frequently than before.
- A sample document has been provided which contains a macro which can be used to paint the empty markers in an assembled document.

2014-04-23 Version 6.0.4

A bug which crept into 6.0.3 (☹) has been fixed. It related to accessing Salesforce data sources.

2014-04-17 Version 6.0.3

A bug which crept into 6.0.2 has been fixed. It related to relevance inside predicates.

2014-04-11 Version 6.0.2

There are a number of bug fixes in this release, pertaining mostly to fairly complex situations as reported by our user base. The users who are affected by them will see that they have been fixed.

One particular user request was that the automatic insertion of a full-stop (period) at the end of the error message in a *Rule* command be removed. This has been done, and it will mean that if the template author really wants a period at the end of a *Rule* error message, this will have to be inserted manually.

2014-03-11 Version 6.0.1

1. Some features have been improved. In particular, the response time when capturing in a grid in the desktop interview has been improved.
2. The *Eq* function has been introduced. This allows wild-cards to be used when comparing long strings. For example, if the data element *MaritalStatus* contains one of the various values for a (South African) marital status, the *If* command
«*If(Eq(MaritalStatus,'mar*com*other'))*» is easier to type and is less prone to error than
«*If(MaritalStatus = 'Married in Community of Property to Each Other')*».

2014-02-21 Version 6.0.0

The most important new feature in Version 6 is the ability, using the data source commands, to create new rows in the underlying data base of the data source. Up to this version, XpressDox has permitted reading of existing data and updating individual columns (fields) in the data. Now it is possible for the template author to instruct XpressDox to create new data rows in the data sources.

1. The *ChooseFromDataSource*, *IncludeDataSourceData*, *LinkToDataSource*, and the new command *LinkToDataSourceOnEnter*, all support the new option called *AllowInsert*. This option appears where *Refresh*, *RefreshSave*, etc. occur. This functionality is covered in the [Cookbook](#).
2. The above commands all now have an option which can be used to govern, on a per-user basis for example, whether the “Save” part of “RefreshSave” will be executed. Thus the same template and data source can be run by different people, and any data changes made by those users in the interview will be saved back to the database only if the particular user is suitably authorized.
3. The *ReadOnly* command can be given a condition which is tested when the interview is constructed, and the data element(s) listed in the command will be read-only only when the condition is true. This condition could, for example, be made to function on a per-user basis, so that some users would be allowed to modify the value of the data element and others would not.
4. A number of bugs which have been found have been fixed. Typically they affected only a few users each, and those users will recognise them from their disappearance.

2014-02-12 Version 5.4.6

1. Template Runners were being prevented from saving their own Home configuration. This is now fixed - a Runner CAN save her/his own Home configuration.

2014-02-05 Version 5.4.5

1. When a template with *IncludeFileData* or *ChooseFromFile* was run, the system would give an error message if the user did not have *Write* permissions on that text file. This is now fixed.

2. When the user presses *Run Template* and the XpressDox Explorer shows the recently used files, the list of recently used files is filtered to include only files with the XpressDox template extensions, not other files.

2014-01-30 Version 5.4.4

1. The toolbar went missing when there was a treeview in the interview. This is fixed.

2014-01-07 Version 5.4.3

1. An error when trying any of the *SetSaved...* commands in the Command Editor has been fixed.
2. The *Left* and *Right* functions will return the left-most and right-most characters, respectively, from a string.
3. The command *InterviewLogo* will enable a logo to be displayed on the interview.

2013-12-17 Version 5.4.2

1. A number of bugs have been fixed, in particular:
 - a. *ChooseFromRepeatingData* for existing data shows the *DisplayElement* when it is not the same as the chosen data element.
 - b. When a database was being updated and there was a column being updated whose name started with the same string as another column name, this gave rise to a database driver error message. This is now fixed.
2. The *InsertTemplate/Picture/FormattedText* commands now correctly apply relevance to any data elements used in their parameters.
3. The XpressDox Explorer adds documents opened directly in Word into its Recently Used list.

2013-11-22 Version 5.4.1

1. An issue was discovered with a *ForEach* which was subject to a condition which included the name of the repeater data element in it. This has now been fixed.
2. The *RefreshOptionalSave* option on data source commands was not performing properly and has been fixed.

2013-11-20 Version 5.4.0

1. *Include/InsertPicture* will change the extents of the image in the document so that the aspect ratio of the actual image is maintained, but the image in the document will remain within the rectangle specified by the command.
2. *Required* is no longer only conditionally relevant, it is also only relevant if the data element is used somewhere else in the template. It means if a template author used the fact that *Required* implied *CaptureDataElement*, then this implication will no longer function.
3. There are some situations where, for example, *If/Else/End* commands need to appear in their own paragraphs. In the past some of these situations were not detected correctly. This has been fixed in this release, but it might mean that templates that appeared to function correctly in the past will now issue an error message relating to commands being required in their own paragraphs.
4. The XpressDox Explorer has a "Check Syntax" option which will allow template authors to select a folder and all the templates in that folder will be tested for syntax correctness.
5. Condition handling inside `<ForEach()>` loops has been tightened up.
6. In Version 5.3.3 the "copy dataset to clipboard" debugging feature was added. This debugging feature has been enhanced in that the internal XML schema governing the interview can also be copied to the clipboard (and then pasted into a text editor for viewing).
7. The *Lookup* function will enable the template author to, for example, select the regional spelling of a word depending on language. For example:

«Lookup(Language,'en-gb:colour;en-us:color')»

2013-10-25 Version 5.3.3

1. A template-author debugging capability is that when you are running an interview (on the desktop) pressing right-click will give the option to copy the data set (i.e. the XML of what's been captured so far) into the clipboard. You can then paste into Notepad or Notepad++ and see the data set.
2. The merge field <TieBreak> can now be used in the configuration of a pattern for the saved document and/or data file name. It will ensure a unique number so that file names can be made unique in ways different to the default XpressDox mechanism.
3. Unbalanced parentheses in “long” text is no longer regarded as an error.
4. The Time Picker control on the desktop has been fixed - it was returning 12:00 AM regardless of what was chosen.
5. The *InsertPicture* command can now take a URL as an argument and not just a file-system file name.

2013-09-13 Version 5.3.2

1. A bug which crept into 5.3.1 regarding the *ChooseFromDataSource* command has been fixed.

2013-09-12 Version 5.3.1

This is a bug-fix release.

1. The PDF form filling utility can be asked to create an interview template. This feature would create an interview with empty captions. This is now fixed, and there will be captions available in the generated interview.
2. There was a problem when XpressDox started up in Word 2013, which showed up particularly with the <<RunAsHugeTemplate()>> command. This has now been fixed.

2013-08-29 Version 5.3.0

1. The *InsertFormattedText* function no longer changes double quotes (“”) into single quotes (”).
2. The database data source search dialog will display the description of the first collection in the header of the form.
3. The relevance engine regarding the treeview for repeaters in the desktop interview has been overhauled.
4. There is a new command to add more functionality to the relevance of repeaters in the interview. <<InterviewRelevance(Name,Name != '')>> will remove the *Name* data element from the interview if that data element is not empty. This is to accommodate the template author when data that the user has not explicitly captured is included in an interview (this can happen, for example, in response to a *ChooseFromDataSource* or *IncludeFileData* or similar command). Up until now, the relevance engine has interpreted conditions such as <<If(Name != '')>>Name is <<Name>><<End()>> saying, in effect “If you want to make the appearance of *Name* in the document depend on its own value, then *Name* will HAVE to be relevant in the interview in order for the user to indicate what its value is”. The *InterviewRelevance* command allows the template author to override this behaviour.
5. The commands *SetSavedDocumentFileName*, *SetSavedDocumentFolder*, *SetSavedDataFileName* and *SetSavedDataFolder* can now take functions as parameters. For example, <<SetSavedDocumentFileName(Doc <AccountNumber>)>> can now be replaced by <<SetSavedDocumentFileName(concat('Doc ', AccountNumber))>>. This is particularly useful if the construction of the document name is fairly complex, and permits constructs such as <<SetSavedDocumentFileName(concat('Letter-',Today('yyyyMMdd'),'-',GetValidFileName(AccountNumber)))>>.

6. In the configuration for a folder, the Helper folders can have <WindowsLogonUser> as part of their path definition.
7. When displaying repeaters in the interview, the *SetRepeaterQualifier* can now be used to display the position of the repeater in the list of repeaters, using the new pseudo-data element *PositionInRepeater*. For example
<<SetRepeaterQualifier(Child,<PositionInRepeater>. <FirstNames>)>>
8. In the data source Search dialog (for databases) it is now possible to sort the collection (table) on a column which is not visible in the dialog.

2013-07-17 Version 5.2.1

1. The Number-to-Words utility has been modified to handle different Regional Settings more effectively.
2. The Clauses utility, which was upgraded in version 5.2.0, has undergone further improvements. In particular, inserted clauses can now have fillpoints in them, and the fillpoints will be completed using the data set that was saved when the merged document was created by XpressDox. If there are “new” fillpoints (i.e. those whose data are not in the data set) then the user will be presented with an interview into which to supply those missing data elements.
3. Bug fixes:
 - a. Table cells which are left empty because of *If* or *Else* commands will now inherit the paragraph formatting of the “nearest” populated cell.
 - b. Regional settings issues with licensing have been resolved.

2013-06-28 Version 5.2.0

1. A long-standing problem has been the error messages that XSLT issues during document assembly time where those messages look something like “Expected token, found ‘.’”, with no context. This could lead to a lot of trouble-shooting effort to find the offending fillpoint. The <<Debug()>> command placed at the top of the template will now detect those syntax errors earlier on and give the template author some context to enable her/him to fix the problem.
2. Fillpoints included into a document in the data (which can happen with data from data sources or included with *IncludeFileData*) are evaluated before the template and data are merged; this is different to how it has been up until now, but it does enable XpressDox to infer correctly when data elements referred to in this way belong to repeaters. It may be that this will not work correctly in some existing scenarios, for example if block commands (e.g. *If* and *ForEach*) have been included in data. Another mechanism will have to be employed by the template author to achieve the same results.
3. If a Word document has been merged with an XpressDox template, and then the user wants to use either the My Clauses or Shared(Office) Clauses utilities to insert clauses into the merged document, then the system will present the user with the XpressDox Explorer, as has always been the case. Prior to this release, the folder at which the Explorer positions itself would be either the Clauses Library Folder defined in the user’s Home configuration, or in the Shared Clauses Library Folder configured by the Supervisor. Now, however, XpressDox will interrogate the active Word document to determine if it was created with an XpressDox template, and, if so, will look in the configuration for the folder from which the template was run, and will use the Clauses Library Folder configured there (or configured in a parent folder). This means that clause libraries can essentially be tailored differently for different sets of templates.
4. Some bugs have been fixed, specifically with regard to the *DefineSetAllGroup*, *RepeatWhile*, and *ChooseFromRepeatingData* commands.
5. A bug involving the selection of folders in the XpressDox Explorer has been fixed.

2013-06-12 Version 5.1.6

This is mainly a bug-fix release, and some new features have been included.

The new features are:

1. The Empty Marker set with *SetEmptyMarker* can be, or start or end with, one or more spaces.
2. The *IncludeFileData* command now supports `<..>` syntax in the construction of the file name, e.g. `<<IncludeFileData(AgreementInfo<AgreementType>.xml)>>` (obviously, as long as the data element name(s) inside the `<..>` are known when the interview is about to be constructed).
3. The inclusion of `<WindowsLogonUser>` (note the single angle brackets) in the configuration for Standard Folders is supported correctly.

Bug fixes include:

1. Extra space after “One” and “Een” in the *NumberPhrase* function is removed.
2. The relevance in the desktop interview of fields not in a grid but depending on values in a grid has been improved.
3. The layout of the first row of a grid where fields in the grid are subject to conditions used to cause the lower part of the first row to be truncated. This is now fixed.
4. Sometimes grids were not being “repainted” when conditions were applied. This is fixed.
5. The default “Documents” folder for the merged documents is no longer created when a merged document is not in fact saved.
6. When the condition on an *OnExistSet* or *OnEnterSet* terminated in multiple right-hand parentheses, these parentheses were deleted. No more.
7. Relevance in the desktop interview is improved where fields are dependent on the *count* function applied to repeaters.

2013-04-30 Version 5.1.4 and 2013-05-01 Version 5.1.5

A bug relating to repeaters disappearing when they are included via *IncludeFileData* or *IncludeDataSourceData* has been fixed.

2013-04-26 Version 5.1.3

This is mainly a bug-fix release, but two new features have also been released.

The new features are:

1. The *ForEvery* command which was released in 5.1.1 has now been deprecated. Instead, the *ForEach* command now uses the ultra-fast mechanism introduced with *ForEvery*. Any usage of *ForEvery* will continue to be supported, but the command will no longer feature in the Command Editor or in the documentation on the web site.
2. In all places where the colour could be specified as an English color name (for Captions, Heading, Footings, etc.), the colour can also be provided as a hexadecimal “RGB” value. For example, `<<Caption(Name,@#FF00BB@The party's name)>>` will color the caption a darkish magenta.

The most important fixes are as follows:

1. With a data source which had name-mapping applied, if the data source was referred to in a *ChooseFromDataSource* with the *Repeat Container* specified (i.e. in order to capture more than one item at time), the name-mapping did not work.
2. There were some problems with refreshing the interview, related to “hidden fields” and also with repeater grids.
3. The presence of a *FormatNumber* function would introduce an automatic *Rule* command, validating the referenced data element as a number. There were complications (the error message would be inappropriate) if a data element was referred to in more than one *FormatNumber*, and also if the template author introduced her/his own *Rule*.

2013-04-12 Version 5.1.1

1. Up until now, the default location for saving the merged document and the captured data set file has been the same folder from which the template was run. This default has now been changed. In the case of the merged document, it is a sub-folder of the template folder called *Documents*, and in the case of the data set it is a sub-folder of the template folder called *Datasets*. If either of these two locations has been configured as some other location, or if another location has been set in the template itself, then this new default will be ignored and that configured or set location will continue to be used as was always the case.
2. In the Desktop, the DatePicker format of the date is dictated by the regional settings, but the format in the XML remains the standard according to the W3C Recommendation (i.e. yyyy-MM-dd).
3. The <<ChooseFromRepeatingData()>> command now has a new option *AddAllChosenData* which adds all the data elements from the chosen repeater into the data set.
4. A new feature is available in the Ribbon-enabled versions of Word (i.e. 2007, 2010 and 2013) called "Re-run template". If a the active Word document has been generated by XpressDox (subsequent to version 5.1.1), then clicking the Re-run Template button will present an interview with the data set which was saved when the document was generated, and will run the template which was originally used to produce that document.
5. A new command - *ForEvery* - has been released, but is not (yet) included in the Command Editor. It is a very fast version of *ForEach* and can be used in most places except nested with other block commands (*If*, *ForEach*, etc.) in a table row.
6. The <<InsertInto(Name,3,'-')>> function will insert the string '-' at position 3 in the value of the data element *Name*.
7. The database data source configuration form stores its size and location for next use.
8. Users not licenced to update configurations can nonetheless view the affected configuration in the configuration user interface, but cannot update them.
9. Bug fixes:
 - a. *ToTitle* capitalises the first character, even if the first word is a short word.
 - b. The database data source configuration form permits deletion of all except 1 first level collection.
 - c. When a new collection is added to a database data source, then the name of the collection is shown in the treeview node for the collection.

2013-03-20 Version 5.1.0

1. It is now possible to run XpressDox templates from within Outlook (2010 and 2013).
2. After this version, the template author will be informed of new version availability automatically.
3. Data sources can be configured so that the template author controls more of the look and feel of the data source search dialog.

2013-03-11 Version 5.0.1

1. The *RepeaterCaption* command will provide a caption for a repeater: in the desktop version of XpressDox this is applicable only to the *CaptureInGrid* situation, whereas in the browser (Cloud) version it will always be applicable.
2. Some bug fixes have been applied:
 - a. A multi-paragraph block (e.g. *If* or *ForEach*) in a table cell (when it was the only such block in the cell) was leaving the cell with no paragraph, rather than an empty paragraph. This is now fixed.

- b. *CaptureAsLongText* was inserting a control on the interview regardless of the relevance rules. A new parameter has been added to enable the author to make the control for *CaptureAsLongText* only display when the data element is relevant.
- c. *AlignCaption* was making the control relevant despite the relevance rules. This has been fixed.

2013-02-28 Version 5.0.0

1. Some new features which have been in the wings for a long time have now been released. In particular, these are the [<<CaptureInGrid\(\)>>](#) and [<<DropDownPanel\(\)>>](#) commands. The first will present a repeater in the interview as a grid, rather than as a node in the interview treeview. And the *DropDownPanel* gives a way, amongst many other things, of presenting a “multiple-choice drop down”.
2. The [<<OnEnterSet\(\)>>](#) command functions in the same way as *OnExitSet*, but triggers its action when the focus enters a control, rather than (as with *OnExitSet*) when it leaves the control.
3. Multiple entries can now be selected in the search dialog presented by the [<<ChooseFromDataSource\(\)>>](#) command. These entries are each added to the dataset as repeated entries. They will appear in the treeview, or, if the *CaptureInGrid* command has been used, in a grid on the interview form.
4. It is now possible to instantiate COM (Common Object Model) and .NET objects and address their methods and properties. There are a huge number of applications which are exposed via the COM mechanism, including all the Microsoft Office modules. These features are discussed in the [Interact with External Programs](#) article.
5. The concept of [Arrays](#) has been introduced. Arrays are similar to variables, and can be manipulated by a set of functions similar to those available for variables. Arrays can be indexed by numeric values (much like VB or arrays in other languages), but also by string values. This makes the array concept very powerful in XpressDox.
6. The [<<MakeNoise\(\)>>](#) function enables the template author to provide audio feedback at different points during the merging of a template.
7. There are so many commands and functions now available that the [Command Editor](#) is becoming cumbersome. To alleviate this, a new presentation format has been added - along with the alphabetic and categorical displays of the commands, a Most Frequently Used display is also available. Initially for any one template author this list is empty, but as the author uses commands or functions they are added into the MFU list and are presented with the most popular commands at the top of the list.
8. The [<<IncrementDate\(\)>>](#) function now has the option to increment by a number of working days. As part of the support for this, a list of public holidays can be provided in the “Other Settings” part of the XpressDox configuration.
9. [<<FlagTOCToBeUpdated\(Yes\)>>](#) has been enhanced to include not only the Word table of contents, but all Word fields.
10. Text files which have XpressDox fillpoints in them can now be “[run](#)” as if they were normal templates. This will help with tasks such as generating web pages and even program code.