XmlDocument 1
XmlElement 3
XmlNodeList 3
XmlNode 4
Streams 8
XmlNamespaceManager 9
Encoding/Convert 9
Http 10
Reimplement codeunit 6224 "XML DOM Management" procedures 11
Record TempBlob to codeunit "Temp Blob" 12
Other breaking changes 14
Codeunit 1 20

XmlDocument

vDotNetXmlDoc: DotNet XmlDocument;	vXmlDocument: XmlDocument;
vDotNetXmlDoc.XmlDocument()	XmlDocument.Create()
vDotNetXmlDoc.LoadXml(*Text*);	XmlDocument.ReadFrom(*Text*, vXmlDocument)
vDotNetXmlDoc.Save(*FileName*)	XmlDocument.WriteTo(*FileName*)
ResponseXMLDocElement := vDotNetXmlDoc.DocumentElement();	vXmlDocument.GetRoot(ResponseXMLDocElement);
vDotNetXmlDoc.Load(*pFileName*);	XmlDocument.ReadFrom(*pFileName*, vXmlDocument);

vDotNetXmlDoc.AppendChild(XmlElement);	vXmlDocument.Add(XmlElement);
vDotNetXmlDoc.CreateXmlDeclaration('1.0', 'UTF-8', '');	XmlDeclaration.Create('1.0', 'UTF-8', '');
vDotNetXmlDoc.InsertBefore(XmlDeclaration, XmlDocument.DocumentElement);	vXmlDocument.AddBeforeSelf(XmlDeclaration, XmlDocument.GetRoot(XmlElement));
vDotNetXmlDoc.DocumentElement();	vXmlDocument.GetRoot(XmlElement);
vDotNetXmlDoc.CreateElement(*pPrefix*, *pName*, *pNamespace*);	XMLDocument.Create(*pPrefix*, *pName*, *pNamespace*).AsXmlNode();

ProcessingInstruction: DotNet XmlProcessingInstruction;
vDotNetXmlDoc := XmlDocument.XmlDocument();
ProcessingInstruction := XmlDoc.CreateProcessingInstruction('xml', 'version="1.0" encoding="UTF-8" standalone="yes"');
CurrNode := vDotNetXmlDoc.AppendChild(ProcessingInstruction);
CurrNode := vDotNetXmlDoc.CreateElement('*soapenv:Envelope*');
CurrNode := vDotNetXmlDoc.AppendChild(CurrNode);
	ProcessingInstruction: XmlProcessingInstruction;
	vXmlDocument := XmlDocument.Create();
	ProcessingInstruction := XmlProcessingInstruction.Create('xml', 'version="1.0" encoding="UTF-8" standalone="yes"');
	CurrNode := ProcessingInstruction.AsXmlNode();
	vXmlDocument.Add(CurrNode);
	Element := XmlElement.Create('*soapenv:Envelope*');
	CurrNode := Element.AsXmlNode();
	vXmlDocument.Add(CurrNode);

vXMLNodelist := vDotNetXmlDoc.GetElementsByTagName('*DataItem*');	vXMLNodelist := vXmlDocument.GetDescendantElements('*DataItem*');

XMLDOMMgt: Codeunit "XML DOM Management";
XMLDOMMgt.AddDeclaration(XmlDoc, '1.0', 'utf-8', 'yes');
	XmlDoc: XmlDocument;
	XmlDoc.SetDeclaration(XMLDeclaration.Create('1.0', 'utf-8', 'yes'));

XmlElement

vDotNetXmlElem: DotNet XmlElement;	vXmlElement: XmlElement;
vDotNetXmlElem.GetElementsByTagName('*Tag*');	vXmlElement.GetDescendantElements('*Tag*');
vDotNetXmlElem.Prefix := v*Text*;	vXmlElement.SetAttribute('Prefix', v*Text*);
vDotNetXmlElem.InnerXml := v*Value*;	vXmlElement.SetAttribute('*InnerXml*', *pText*);
vParentNode.AppendChild(vDotNetXmlElem);	vParentNode.AddAfterSelf(vXmlElement);
vDotNetXmlElem.GetElementsByTagName('*return_value*');	vXmlElement.GetDescendantElements('*return_value*');

XmlNodeList

vDotNetXmlNodeList: DotNet XmlNodeList;	vXmlNodeList: XmlNodeList;

Text := vDotNetXmlNodeList.Item(0).InnerXml();	vXmlNodeList.Get(0, vXMLNode);
	Text := vXMLNode.AsXmlElement().InnerXml();

vDotNetXMLNode := vDotNetXmlNodeList.Item(**i**);	vXmlNodeList.Get(**i**, vXMLNode);

Text := vDotNetXmlNodeList.Item(0).InnerXml();	vXmlNodeList.Get(0, vXMLNode);
	Text := vXMLNode.AsXmlElement().InnerXml();

Integer := vDotNetXmlNodeList.Item(0).ChildNodes.Count;	vXmlNodeList.Get(0, vXMLNode);
	Integer := vXMLNode.AsXmlElement().GetChildNodes().Count;

XmlNode

vDotNetXmlNode: DotNet XmlNode;	vXmlNode: XmlNode;
vDotNetXmlNode.InnerText;	vXmlNode.AsXmlElement().InnerText();
vDotNetXmlNode.Name;	vXmlNode.AsXmlElement().Name;

IsNull(vDotNetXmlNode);	vXmlNode.AsXmlElement().GetChildElements().Count = 0;
	or
	vXmlNode.AsXmlElement().IsEmpty;

vDotNetXmlNode.OuterXml;	vXmlNode.AsXmlElement().InnerText();

text := DelChr(vDotNetXmlNode.Attributes().GetNamedItem('*category*').Value, '<>', ' ');	vXmlAttribute: XmlAttribute;
	vXmlNode.AsXmlElement().Attributes().Get('*category*', vXmlAttribute);
	text := DelChr(vXmlAttribute.Name, '<>', ' ');

vDotNetXmlNode.ChildNodes.Count ;	vXmlNode.AsXmlElement().GetChildElements().Count;

Text := DelChr(vDotNetXmlNode.AsXmlDocument().GetChildElements().Item(i) .Attributes().GetNamedItem('*country*').InnerText, '<>', ' ');	vXmlAttribute: XmlAttribute;
	vXmlNode.AsXmlElement().GetChildElements().Get(i, vXmlNode2);
	vXmlNode2.AsXmlElement().Attributes().Get('*country*', vXmlAttribute);
	Text := DelChr(vXmlAttribute.AsXmlNode().AsXmlElement().InnerText, '<>', ' ');

vDotNetXmlNode := vDotNetXmlNode2.SelectSingleNode('*details/further_information*');	vXmlNode2.SelectSingleNode('*details/further_information*', pXmlNode);

Text := vDotNetXmlNode.SelectNodes('*FROMJURI*').Item(0).InnerText;	vXmlNode.SelectNodes('*FROMJURI*', vXmlNodeList);
	vXmlNodeList.Get(0, vXmlNode);
	Text := vXmlNode.AsXmlElement().InnerText();

Evaluate(_Integer, vDotNetXmlNode.SelectNodes('*RECORDCOUNT*').Item(0).InnerText);	vXmlNode.SelectNodes('*RECORDCOUNT*', vXmlNodeList);
	vXmlNodeList.Get(0, vXmlNode);
	Evaluate(_Integer, vXmlNode.AsXmlElement().InnerText);

CreateNode(vXMLDocument.DocumentElement, Text);	vXMLDocument.AsXmlNode().AsXmlDocument().GetRoot(vXmlElement);
	CreateNode(vXmlElement, Text);

foreach XMLNode in vDotNetXmlNodedo begin	vXMLNodeList := vXMLNode.AsXmlElement().GetChildElements();
	
	foreach XMLNode in vXMLNodeList do begin

vXMLElement := vDotNetXmlNodeList.Item(i);	vXMLNodeList .Get(i, vXmlNode);
	vXMLElement := vXmlNode.AsXmlElement();

vDotNetXMLNode := vDotNetXmlNodeList.Item(0);	vXmlNodeList.Get(0, vXmlNode);
vDotNetXMLNode.AsXmlElement().InnerText() := Text;	vXmlNode := XmlElement.Create(NodeName, NameSpace, Text).AsXmlNode()
if IsNull(vXMLNode) then	if vXmlNode.AsXmlElement().IsEmpty then

vDotNetXMLNode := vDotNetXmlNodeList.Item(0).ChildNodes.Item(0);	vXMLNodeList.Get(0, vXmlNode);
	
	vXMLNodeList:= vXmlNode.AsXmlElement().GetChildNodes();
	vXMLNodeList.Get(0, vXmlNode);

Text := vDotNetXMLNode.Attributes.Item(1).InnerText;	vXmlNode.AsXmlElement().Attributes.Get(1, vXmlAttribute);
	Text := vXmlAttribute.AsXmlNode().AsXmlElement().InnerText;

vDotNetXMLNode.OwnerDocument	vXmlNode.GetDocument

vDotNetXMLNode := XmlNode.FirstChild;	vXMLNodeList := XMLNode.AsXmlElement().GetChildElements();
	vXMLNodeList.Get(0, XMLChildNode);

XMLNode.Attributes.SetNamedItem(XMLNewAttributeNode);	astAttributeIndex := XMLNode.AsXmlElement().Attributes.Count();
	XMLNode.AsXmlElement().Attributes.Get(LastAttributeIndex - 1, LocalXmlAttribute);
	LocalXmlAttribute.AddAfterSelf(XMLNewAttributeNode);

XMLNewAttributeNode := vDotNetXMLNode.OwnerDocument.CreateAttribute(Name);	XMLNode.GetDocument(localXmlDocument);
	XMLNewAttributeNode := XmlAttribute.Create(Name, Name).AsXmlNode();

vDotNetXMLNode.Prefix;	XMLNode.AsXmlAttribute().NamespacePrefix;
vDotNetXMLNode.Name;	XMLNode.AsXmlAttribute().Name;
vDotNetXMLNode.NamespaceURI;	XMLRootNode.AsXmlElement().NamespaceUri;

vDotNetSiblingXmlNode := vDotNetXmlNode.NextSibling;	local procedure NextSibling(CurrNode: XmlNode): XmlNode
	var
	 ParentEl: XmlElement;
	 vXmlNodeList: XmlNodeList;
	 vXmlAttributeParameter: XmlAttribute;
	 NodeCount: Integer;
	 i: Integer;
	 XmlNodeLocal: XmlNode;
	 vXmlAttributeLocal: XmlAttribute;
	 NextNode: XmlNode;
	begin
	 CurrNode.GetParent(ParentEl);
	 vXmlNodeList := ParentEl.GetChildNodes();
	 vXmlAttributeParameter := CurrNode.AsXmlAttribute();
	 NodeCount := vXmlNodeList.Count();
	 for i := 0 to NodeCount - 1 do begin
	 vXmlNodeList.Get(i, XmlNodeLocal);
	 vXmlAttributeLocal := XmlNodeLocal.AsXmlAttribute();
	 if (vXmlAttributeLocal.Value = vXmlAttributeParameter.Value) then begin
	 i := i + 1;
	 break;
	 end;
	 end;
	
	 vXmlNodeList.Get(i, NextNode);
	 exit(NextNode);
	
	end;

vDotNetXmlChildNode :=
vDotNetXmlDoc.RemoveChild(vDotNetXmlChildNode);
	ChildNodes: XmlNodeList;
	Xml_Element: XmlElement;
	Xml_Attribute: XmlAttribute;
	XmlAttributeParameter: XmlAttribute;
	NodeCount: Integer;
	i: Integer;
	Xml_Node: XmlNode;
	
	ChildNodes := Xml_Element.GetChildNodes();
	        XmlAttributeParameter := XMLChildNode.AsXmlAttribute();

	NodeCount := ChildNodes.Count();
	for i := 0 to NodeCount - 1 do begin
	    ChildNodes.Get(i, Xml_Node);
	    Xml_Attribute := Xml_Node.AsXmlAttribute();
	    if (Xml_Attribute.Value = XmlAttributeParameter.Value) then begin
	        Xml_Node.Remove();
	        break;
	    end;
	end;

FieldNode.ParentNode.InsertAfter(NewDoc, FieldNode.NextSibling);	FieldNode.GetParent(ParentEl);
	NextXmlNode := NextSibling(FieldNode);
	NextXmlNode.AddAfterSelf(NewDoc);

Streams

vDotNetXMLDocument: DotNet XmlDocument;
vDotNetXMLDocument.Save(FullFilePath);
	IStream: InStream;
	
	OStream: OutStream;
	
	TempBlob: Codeunit "Temp Blob";
	
	vXMLDocument: XmlDocument;ReadText
	
	TempBlob.CreateOutStream(OStream);
	
	vXMLDocument.WriteTo(OStream);
	TempBlob.CreateInStream(IStream);

Download(FullFilePath, *'Export'*, '', *FilterString*, *ToFile*);	DownloadFromStream(IStream, *'Export'*, '', *FilterString*, *ToFile*)

XmlNamespaceManager

XmlNamespaceMng: DotNet XmlNamespaceManager;
 	
XmlNamespaceMng := XmlNamespaceMng.XmlNamespaceManager(XmlDoc.NameTable);
XmlNamespaceMng.AddNamespace('SOAP-ENV', 'http://schemas.xmlsoap.org/soap/envelope/');
	XmlNamespaceMng: XmlNamespaceManager;
	
	XmlNamespaceMng.NameTable(XmlDoc.NameTable);
	XmlNamespaceMng.AddNamespace('SOAP-ENV', 'http://schemas.xmlsoap.org/soap/envelope/');

Encoding/Convert

SystemUTF8Encoder: DotNet UTF8Encoding;
 	
SystemUTF8Encoder := SystemUTF8Encoder.UTF8Encoding;
SystemByteArray := SystemUTF8Encoder.GetBytes(s);
	SystemByteArray: Text;
	Base64Convert: Codeunit "Base64 Convert";
	
	SystemByteArray := Base64Convert.FromBase64(s);

Encoding: DotNet Encoding;
MessageBytes: DotNet Array;
 	
MessageBytes := Encoding.UTF8.GetBytes(TxtToHash);
	MessageBytes: List of [Text];
	Convert: Codeunit "Base64 Convert";
	
	MessageBytes.Add(Convert.ToBase64(TxtToHash));

Http

HttpWebRequest: DotNet HttpWebRequest;
HttpWebResponse: DotNet WebResponse;
StreamReader: DotNet StreamReader;
RequestStream: DotNet Stream;
 	
HttpWebRequest := HttpWebRequest.Create(Url);
HttpWebRequest.Method := 'POST';
HttpWebRequest.Timeout := 30000;
HttpWebRequest.ContentType := 'application/soap+xml;charset=UTF-8';
XmlDoc.WriteTo(XmlDocText);
ByteArray := Encoding.UTF8.GetBytes(XmlDocText);
HttpWebRequest.ContentLength := ByteArray.Length;
RequestStream := HttpWebRequest.GetRequestStream;
RequestStream.Write(ByteArray, 0, ByteArray.Length);
RequestStream.Close;
HttpWebResponse := HttpWebRequest.GetResponse;
StreamReader := StreamReader.StreamReader(HttpWebResponse.GetResponseStream);
ResponseText := StreamReader.ReadToEnd;
	ByteArray: Text;
	HttpWebRequest: HttpRequestMessage;
	RequestText: Text;
	HttpWebResponse: HttpResponseMessage;
	WebContent: HttpContent;
	WebHeader: HttpHeaders;
	WebClient: HttpClient;
	Convert: Codeunit "Base64 Convert";
	
	XmlDoc.WriteTo(XmlDocText);
	WebContent.WriteFrom(Convert.ToBase64(XmlDocText));
	WebContent.GetHeaders(WebHeader);
	WebHeader.Clear();
	WebHeader.Add('SOAPAction', 'urn:microsoft-dynamics-schemas/codeunit/PortalService:GetUpdatePortalMessagesWS');
	HttpWebRequest.Content(WebContent);
	HttpWebRequest.SetRequestUri(Url);
	HttpWebRequest.Method := 'POST';
	WebClient.Clear();
	WebClient.Send(HttpWebRequest, HttpWebResponse);
	WebClient.Timeout(30000);
	HttpWebResponse.Content.ReadAs(ResponseText);

HttpWebRequest := HttpWebRequest.Create(ServiceUrl);
HttpWebRequest.Method := 'POST';
HttpWebRequest.KeepAlive := true;
HttpWebRequest.AllowAutoRedirect := true;
HttpWebRequest.Headers.Add('SOAPAction', SoapAction);
HttpWebRequest.ContentType := 'text/xml;charset=utf-8';
HttpWebRequest.Accept := 'text/xml';
HttpWebRequest.Timeout := 600000;
HttpWebRequest.AutomaticDecompression := DecompressionMethods.GZip;
	HttpWebRequest.SetRequestUri(ServiceUrl);
	HttpWebRequest.Method('POST');
	HttpWebRequest.GetHeaders(Headers);
	Headers.Add('KeepAlive', 'true');
	Headers.Add('AllowAutoRedirect', 'true');
	Headers.Add('ContentType', 'text/xml;charset=utf-8');
	Headers.Add('Accept', 'text/xml');
	Headers.Add('Timeout', '600000');
	Headers.Add('AutomaticDecompression', 'GZip');
	Headers.Add('SOAPAction', SoapAction);

if not WebRequestHelper.GetWebResponse(HttpWebRequest, HttpWebResponse, ResponseInStream, HttpStatusCode, ResponseHeaders, false) then	if not Http.Send(HttpWebRequest, HttpWebResponse) then

HttpWebResponse.GetResponseStream.CopyTo(ResponseInStream);
IStream.ReadText(ResponseText);
	HttpWebResponse.Content.ReadAs(ResponseText);

DotNet_HttpWebRequest.Credentials := DotNet_CredentialCache;	HttpWebRequest.GetHeaders(Headers);
	Headers.Add('Credentials', DotNet_CredentialCache.ToString());

DotNet_HttpWebRequest.Headers.Add('SOAPAction', SoapAction);	HttpWebRequest.GetHeaders(Headers);
	Headers.Add('SOAPAction', SoapAction);

HttpWebResponse := HttpWebRequest.GetResponse;	Http.Send(HttpWebRequest, HttpWebResponse);
	HttpWebResponse.Content.ReadAs(Vehicles);

HttpWebRequest := HttpWebRequest.Create(StrSubstNo(Url, Action));	HttpWebRequest.SetRequestUri(StrSubstNo(Url, Action));

Reimplement codeunit 6224 "XML DOM Management" procedures

XMLElementArray: array[100] of DotNet XmlNode;
XMLDOMManagement.SaveXMLDocumentToOutStream(FileOutStream, XMLElementArray[1]);
	XMLElementArray: array[100] of XmlNode;
	XMLElementArray[1].GetDocument(XmlDoc);
	XmlDoc.WriteTo(FileOutStream);

vDotNetXmlDocument: DotNet XmlDocument;
XmlStream: InStream;
XMLDOMManagement.LoadXMLDocumentFromInStream(XmlStream, vDotNetXmlDocument);
	vXmlDocument: XmlDocument;
	XmlStream: InStream;
	XmlDocument.ReadFrom(XmlStream, vXmlDocument);
	vXmlDocument.GetRoot(vXmlElement);

XMLDOMManagement.FindNodeXML(vDotNetXmlDocument, 'type');	local procedure FindNodeXML(vXMLElement: XmlElement; NodePath: Text): Text
	var
	vXMLNode: XmlNode;
	begin
	if vXMLElement.SelectSingleNode(NodePath, vXMLNode) then
	exit(vXMLNode.AsXmlElement().InnerXml)

	else
	
	exit('');
	end;

XMLDOMManagement.AddDeclaration(vDotNetXMLDocument, *'1.0'*, *'UTF-8'*, *''*);	Declaration: XmlDeclaration;
	
	Declaration := XmlDeclaration.Create(*'1.0'*, *'UTF-8'*, *''*);
	vXMLDocument.AddFirst(Declaration);

Record TempBlob to codeunit "Temp Blob"

TempBlobRecord.Init();
 	
TempBlobRecord.Blob := VATReportArchive."Message BLOB";
	Clear(TempBlobCodeunit);
	
	TempBlobCodeunit.FromRecord(VATReportArchive, VATReportArchive.FieldNo("Message BLOB"));

TempBlobRecord.CalcFields(Blob);
 	
VATReportArchive."Message BLOB" := TempBlobRecord.Blob;
	var
	  RecRef: RecordRef;
	
	RecRef.GetTable(VATReportArchive);
	
	TempBlobCodeunit.ToRecordRef(RecRef, VATReportArchive.FieldNo("Message BLOB"));
	
	RecRef.SetTable(VATReportArchive);

ContentTxt := TempBlobRecord.ReadAsText('', TextEncoding::UTF8);	var
	  IStream: InStream;
	TempBlobCodeunit.CreateInStream(IStream, TextEncoding::UTF8);
	IStream.ReadText(ContentTxt);

TempBlobRecord.WriteAsText(ContentTxt, TextEncoding::UTF8);	var
	  OStream: OutStream;
	TempBlobCodeunit.CreateOutStream(OStream, TextEncoding::UTF8);
	OStream.WriteText(ContentTxt);

TempBlobRecord2.CalcFields(Blob);
TempBlobRecord1.Blob := TempBlobRecord2.Blob;
	var
	  OStream: OutStream;
	  IStream: InStream;
	TempBlobCodeunit1.CreateOutStream(OStream);
	TempBlobCodeunit2.CreateInStream(IStream);
	CopyStream(OStream, IStream);

In Reports:
TempBlobRecord.Blob := StatReportSetup."Tick Box Filled";
column(Blob; TempBlobRecord.Blob)
{
}
	var
	  TenantMediaTemp: Record "Tenant Media" temporary;
	TenantMediaTemp.Content:= StatReportSetup."Tick Box Filled";
	column(Blob; TenantMediaTemp.Content)
	{
	}

TempBlobRecord.Init();
TempBlobRecord.FromBase64String(base64encodeString);
	var
	Base64Convert: Codeunit "Base64 Convert";
	OStream: OutStream;
	TempBlobCodeunit.CreateOutStream(OStream);
	OStream.WriteText(Base64Convert.FromBase64(base64encodeString));

TempBlobRecord.Init();
TempBlobRecord.Blob := "Portal Files"."The File";
TempBlobRecord.CalcFields(Blob);
base64encodeString := TempBlobRecord.ToBase64String();
	var
	Base64Convert: Codeunit "Base64 Convert";
	IStream: InStream;
	TempBlobCodeunit.FromRecord("Portal Files", "Portal Files".FieldNo("The File"));
	TempBlobCodeunit.CreateInStream(IStream);
	IStream.ReadText(SimpleText);
	base64encodeString := Base64Convert.ToBase64(SimpleText);

Other breaking changes

Object	**Obsolete procedure**	**Workaround**
Table 37 “Sales Line”	CrossReferenceNoLookUp	ItemReferenceMgt.SalesReferenceNoLookup(Rec);
Table 77 “Report Selections”	FindPrintUsage	FindReportUsageForCust
Table 77 “Report Selections”	FindPrintUsageVendor	FindReportUsageForVend

Table 77 “Report Selections”	FindEmailAttachmentUsage	FindEmailAttachmentUsageForCust
Table 77 “Report Selections”	FindEmailAttachmentUsageVendor	FindEmailAttachmentUsageForVend
Table 77 “Report Selections”	FilterEmailUsage	SetEmailUsageFilters
Table 77 “Report Selections”	FilterEmailBodyUsage	SetEmailBodyUsageFilters
Table 77 “Report Selections”	FilterEmailAttachmentUsage	SetEmailAttachmentUsageFilters
Table 77 “Report Selections”	FindEmailBodyUsage	FindEmailBodyUsageForCust
Table 77 “Report Selections”	FindEmailBodyUsageVendor	FindEmailBodyUsageForVend

Table 77 “Report Selections ”	PrintWithCheck	PrintWithCheckForCust
Table 77 “Report Selections ”	PrintWithGUIYesNoWithCheck	PrintWithDialogWithCheckForCust
Table 77 “Report Selections ”	PrintWithGUIYesNoWithCheckVendor	PrintWithDialogWithCheckForVend
Table 77 “Report Selections ”	PrintWithGUIYesNo	PrintWithDialogForCust
Table 77 “Report Selections ”	PrintWithGUIYesNoVendor	PrintWithDialogForVend
Table 77 “Report Selections ”	GetHtmlReport	GetHtmlReportForCust
Table 77 “Report Selections ”	GetPdfReport	GetPdfReportForCust

Table 77 “Report Selections”	GetEmailBody	GetEmailBodyForCust
Table 77 “Report Selections”	GetEmailBodyCustomText	GetEmailBodyTextForCust
Table 77 “Report Selections”	GetEmailBodyVendor	GetEmailBodyForVend
Table 870 “Social Listening Setup”		Discontinued
Table 871 “Social Listening Search Topic”		Discontinued
Table 5717 “Item Cross Reference”		Use table “Item Reference”
Table 9650 “Custom		Discontinued

Report Layout”		
Page 870 “Social Listening Setup”		Discontinued
Page 871 “Social Listening Search Topic”		Discontinued
Page 875 “Social Listening FactBox”		Discontinued
Page 876 “Social Listening Setup FactBox”		Discontinued
Page 5721 “Item Cross Reference Entries”		Use Page “Item Reference Entries”
Page 9080 “Sales Hist.	SetCustomerNoVisibility	Discontinued, but can be copied to page extension or reimplement to set visibility

Sell-to FactBox”		
Page 9081 “Sales Hist. Bill-to FactBox”	SetCustomerNoVisibility	Discontinued, but can be copied to page extension or reimplement to set visibility
Page 9082 “Customer Statistics FactBox”	SetCustomerNoVisibility	Discontinued, but can be copied to page extension or reimplement to set visibility
Page 9084 “Customer Details FactBox”	SetCustomerNoVisibility	Discontinued, but can be copied to page extension or reimplement to set visibility
Codeunit 871 “Social Listening Management”		Discontinued
Codeunit 6620 “Copy Document Mgt.”	CopyPurchLine	CopyPurchDocLine
Codeunit 6620	CopySalesLine	CopySalesDocLine

“Copy Document Mgt.”		
Codeunit 6620 “Copy Document Mgt.”	PurchHeaderDocType	GetPurchaseDocumentType
Codeunit 6620 “Copy Document Mgt.”	ArchPurchHeaderDocType	GetPurchaseDocumentType
Codeunit 6620 “Copy Document Mgt.”	SalesHeaderDocType	GetSalesDocumentType
Codeunit 6620 “Copy Document Mgt.”	ArchSalesHeaderDocType	GetSalesDocumentType

Codeunit 1

Codeunit 1 trigger	New codeunit ID	New codeunit name	New method
CompanyClose	40	LogInManagement	CompanyClose
CompanyOpen	40	LogInManagement	CompanyOpen
GetSystemIndicator	40	LogInManagement	GetSystemIndicator
OnAfterCompanyClose	40	LogInManagement	OnAfterCompanyClose
OnAfterCompanyOpen	40	LogInManagement	OnAfterCompanyOpen
OnBeforeCompanyClose	40	LogInManagement	OnBeforeCompanyClose
OnBeforeCompanyOpen	40	LogInManagement	OnBeforeCompanyOpen
FindPrinter	44	ReportManagement	GetPrinterName
ApplicationVersion	9015	Application System Constants	ApplicationVersion
CustomApplicationVersion	N/A	N/A	
ReleaseVersion	9015	Application System Constants	ReleaseVersion
ApplicationBuild	9015	Application System Constants	ApplicationBuild
CustomApplicationBuild	N/A	N/A	
ApplicationLanguage	43	LanguageManagement	ApplicationLanguage
DefaultRoleCenter	9170	Conf./Personalization Mgt.	DefaultRoleCenterID
MakeDateTimeText	41	TextManagement	MakeDateTimeText
GetSeparateDateTime	41	TextManagement	GetSeparateDateTime
MakeDateText	41	TextManagement	MakeDateText
MakeTimeText	41	TextManagement	MakeTimeText

Codeunit 1 trigger	New codeunit ID	New codeunit name	New method
MakeText	41	TextManagement	MakeText
MakeDateTimeFilter	41	TextManagement	MakeDateTimeFilter
MakeDateFilter	41	TextManagement	MakeDateFilter
MakeTextFilter	41	TextManagement	MakeTextFilter
MakeCodeFilter	41	TextManagement	MakeTextFilter
MakeTimeFilter	41	TextManagement	MakeTimeFilter
AutoFormatTranslate	45	AutoFormatManagement	AutoFormatTranslate
ReadRounding	45	AutoFormatManagement	ReadRounding
CaptionClassTranslate	42	CaptionManagement	CaptionClassTranslate
GetCueStyle	9701	Cue Setup	GetCueStyle
SetGlobalLanguage	43	LanguageManagement	SetGlobalLanguage
ValidateApplicationlLanguage	43	LanguageManagement	ValidateApplicationLanguage
LookupApplicationlLanguage	43	LanguageManagement	LookupApplicationLanguage
GetGlobalTableTriggerMask	49	GetGlobalTableTriggerMask	
OnGlobalInsert	49	GlobalTriggerManagement	OnGlobalInsert
OnGlobalModify	49	GlobalTriggerManagement	OnGlobalModify
OnGlobalDelete	49	GlobalTriggerManagement	OnGlobalDelete
OnGlobalRename	49	GlobalTriggerManagement	OnGlobalRename
GetDatabaseTableTriggerSetup	49	GlobalTriggerManagement	GetDatabaseTableTriggerSetup

Codeunit 1 trigger	New codeunit ID	New codeunit name	New method
OnDatabaseInsert	49	GlobalTriggerManagement	OnDatabaseInsert
OnDatabaseModify	49	GlobalTriggerManagement	OnDatabaseModify
OnDatabaseDelete	49	GlobalTriggerManagement	OnDatabaseDelete
OnDatabaseRename	49	GlobalTriggerManagement	OnDatabaseRename
OnDebuggerBreak	N/A	N/A	
LaunchDebugger	N/A	N/A	
OpenSettings	9170	Conf./Personalization Mgt.	OpenSettings
OpenContactMSSales	50	SaaS Log In Management	OpenContactMSSales
InvokeExtensionInstallation	2501	ExtensionMarketplaceMgmt	InvokeExtensionInstallation
CustomizeChart	9180	Generic Chart Mgt	CustomizeChart
HasCustomLayout	44	ReportManagement	HasCustomLayout
MergeDocument	44	ReportManagement	MergeDocument
ReportGetCustomRdlc	44	ReportManagement	ReportGetCustomRdlc
ReportScheduler	44	ReportManagement	ScheduleReport
OnBeforeOpenSettings	9170	Conf./Personalization Mgt.	OnBeforeOpenSettings
OnAfterGetApplicationVersion	9015	Application System Constants	OnAfterGetApplicationVersion
OnRoleCenterOpen	9170	Conf./Personalization Mgt.	OnRoleCenterOpen
OnAfterGetSystemIndicator	79	OnAfterGetSystemIndicator	
OnAfterFindPrinter	44	ReportManagement	OnAfterGetPrinterName

Codeunit 1 trigger	New codeunit ID	New codeunit name	New method
OnAfterGetDefaultRoleCenter	9170	Conf./Personalization Mgt.	OnAfterGetDefaultRoleCenter
OnAfterMakeDateText	N/A	N/A	
OnAfterMakeTimeText	N/A	N/A	
OnAfterMakeText	N/A	N/A	
OnAfterMakeDateTimeFilter	41	LanguageManagement	OnAfterMakeDateTimeFilter
OnAfterMakeDateFilter	41	LanguageManagement	OnAfterMakeDateFilter
OnAfterMakeTextFilter	41	LanguageManagement	OnAfterMakeTextFilter
OnAfterMakeCodeFilter	N/A	N/A	
OnAfterMakeTimeFilter	41	LanguageManagement	OnAfterMakeTimeFilter
OnAfterAutoFormatTranslate	45	AutoFormatManagement	OnAfterAutoFormatTranslate
OnAfterCaptionClassTranslate	42	CaptionManagement	OnAfterCaptionClassTranslate
OnAfterGetGlobalTableTriggerMask	49	GlobalTriggerManagement	OnAfterGetGlobalTableTriggerMask
OnAfterOnGlobalInsert	49	GlobalTriggerManagement	OnAfterOnGlobalInsert
OnAfterOnGlobalModify	49	GlobalTriggerManagement	OnAfterOnGlobalModify
OnAfterOnGlobalDelete	49	GlobalTriggerManagement	OnAfterOnGlobalDelete
OnAfterOnGlobalRename	49	GlobalTriggerManagement	OnAfterOnGlobalRename
OnAfterGetDatabaseTableTriggerSetup	49	GlobalTriggerManagement	OnAfterGetDatabaseTableTriggerSetup
OnAfterOnDatabaseInsert	49	GlobalTriggerManagement	OnAfterOnDatabaseInsert
OnAfterOnDatabaseModify	49	GlobalTriggerManagement	OnAfterOnDatabaseModify

Codeunit 1 trigger	New codeunit ID	New codeunit name	New method
OnAfterOnDatabaseDelete	49	GlobalTriggerManagement	OnAfterOnDatabaseDelete
OnAfterOnDatabaseRename	49	GlobalTriggerManagement	OnAfterOnDatabaseRename
OnAfterHasCustomLayout	44	ReportManagement	OnAfterHasCustomLayout
OnAfterReportGetCustomRdlc	9650	Edit MS Word Report Layout	OnAfterReportGetCustomRdlc
OnBeforeOnDatabaseInsert	49	GlobalTriggerManagement	OnBeforeOnDatabaseInsert
OnBeforeOnDatabaseModify	49	GlobalTriggerManagement	OnBeforeOnDatabaseModify
OnBeforeOnDatabaseDelete	49	GlobalTriggerManagement	OnBeforeOnDatabaseDelete
OnBeforeOnDatabaseRename	49	GlobalTriggerManagement	OnBeforeOnDatabaseRename
OnEditInExcel	6710	OnEditInExcel	
OnInstallAppPerDatabase	N/A	N/A	
OnInstallAppPerCompany	N/A	N/A	
OnCheckPreconditionsPerDatabase	9900	OnCheckPreconditionsPerDatabase	
OnCheckPreconditionsPerCompany	9900	RaiseOnCheckPreconditionsPerCompany	
OnUpgradePerDatabase	9900	OnUpgradePerDatabase	
OnUpgradePerCompany	9900	OnUpgradePerCompany	
OnValidateUpgradePerDatabase	9900	OnValidateUpgradePerDatabase	
OnValidateUpgradePerCompany	9900	OnValidateUpgradePerCompany	