



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	21

XmlDocument

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



<pre>vDotNetXmlDoc.AppendChild(XmlElement);</pre>	<pre>vXmlDocument.Add(XmlElement);</pre>
<pre>vDotNetXmlDoc.CreateXmlDeclaration('1.0', 'UTF-8', "");</pre>	<pre>XmlDeclaration.Create('1.0', 'UTF-8', "");</pre>
<pre>vDotNetXmlDoc.InsertBefore(XmlDeclaration, XmlDocument.DocumentElement);</pre>	<pre>vXmlDocument.AddBeforeSelf(XmlDeclaration, XmlDocument.GetRoot(XmlElement));</pre>
<pre>vDotNetXmlDoc.DocumentElement();</pre>	<pre>vXmlDocument.GetRoot(XmlElement);</pre>
<pre>vDotNetXmlDoc.CreateElement(<i>pPrefix, pName, pNamespace</i>);</pre>	<pre>XMLDocument.Create(<i>pPrefix, pName, pNamespace</i>).AsXmlNode();</pre>
<pre>ProcessingInstruction: DotNet XmlProcessingInstruction;</pre> <pre>vDotNetXmlDoc := XmlDocument.XmlDocument();</pre> <pre>ProcessingInstruction := XmlDoc.CreateProcessingInstruction('xml', 'version="1.0" encoding="UTF-8" standalone="yes"');</pre> <pre>CurrNode := vDotNetXmlDoc.AppendChild(ProcessingInstruction);</pre> <pre>CurrNode := vDotNetXmlDoc.CreateElement('soapenv:Envelope');</pre> <pre>CurrNode := vDotNetXmlDoc.AppendChild(CurrNode);</pre>	<pre>ProcessingInstruction: XmlProcessingInstruction;</pre> <pre>vXmlDocument := XmlDocument.Create();</pre> <pre>ProcessingInstruction := XmlProcessingInstruction.Create('xml', 'version="1.0" encoding="UTF-8" standalone="yes"');</pre> <pre>CurrNode := ProcessingInstruction.AsXmlNode();</pre> <pre>vXmlDocument.Add(CurrNode);</pre> <pre>Element := XmlElement.Create('soapenv:Envelope');</pre> <pre>CurrNode := Element.AsXmlNode();</pre> <pre>vXmlDocument.Add(CurrNode);</pre>
<pre>vXMLNodelist := vDotNetXmlDoc.GetElementsByTagName('DataItem');</pre>	<pre>vXMLNodelist := vXmlDocument.GetDescendantElements('DataItem');</pre>
<pre>XMLDOMMgt: Codeunit "XML DOM Management";</pre> <pre>XMLDOMMgt.AddDeclaration(XmlDoc, '1.0', 'utf-8', 'yes');</pre>	<pre>XmlDoc: XmlDocument;</pre> <pre>XmlDoc.SetDeclaration(XmlDeclaration.Create('1.0', 'utf-8', 'yes'));</pre>

XmlElement

vDotNetXmlElem: DotNet XmlElement;	vXmlElement: XmlElement;
vDotNetXmlElem.GetElementsByTagName('Tag');	vXmlElement.GetDescendantElements('Tag');
vDotNetXmlElem.Prefix := vText;	vXmlElement.SetAttribute('Prefix', vText);
vDotNetXmlElem.InnerXml := vValue;	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, '<>', ' ');



<pre>vDotNetXmlNode := vDotNetXmlNode2.SelectSingleNode('details/further_information');</pre>	<pre>vXmlNode2.SelectSingleNode('details/further_information', pXmlNode);</pre>
<pre>Text := vDotNetXmlNode.SelectNodes('FROMJURI').Item(0).InnerText;</pre>	<pre>vXmlNode.SelectNodes('FROMJURI', vXmlNodeList); vXmlNodeList.Get(0, vXmlNode); Text := vXmlNode.AsXmlElement().InnerText();</pre>
<pre>Evaluate(_ Integer, vDotNetXmlNode.SelectNodes('RECORDCOUNT').Item(0).InnerText);</pre>	<pre>vXmlNode.SelectNodes('RECORDCOUNT', vXmlNodeList); vXmlNodeList.Get(0, vXmlNode); Evaluate(_ Integer, vXmlNode.AsXmlElement().InnerText);</pre>
<pre>CreateNode(vXMLDocument.DocumentElement, Text);</pre>	<pre>vXMLDocument.AsXmlNode().AsXmlDocument().GetRoot(vXmlElement); CreateNode(vXmlElement, Text);</pre>
<pre>foreach XmlNode in vDotNetXmlNode do begin</pre>	<pre>vXMLNodeList := vXMLNode.AsXmlElement().GetChildElements(); foreach XmlNode in vXMLNodeList do begin</pre>
<pre>vXMLElement := vDotNetXmlNodeList.Item(i);</pre>	<pre>vXMLNodeList .Get(i, vXmlNode); vXMLElement := vXmlNode.AsXmlElement();</pre>
<pre>vDotNetXMLNode := vDotNetXmlNodeList.Item(0);</pre>	<pre>vXmlNodeList.Get(0, vXmlNode);</pre>
<pre>vDotNetXMLNode.AsXmlElement().InnerText() := Text;</pre>	<pre>vXmlNode := XmlElement.Create(NodeName, Namespace, Text).AsXmlNode()</pre>
<pre>if IsNull(vXMLNode) then</pre>	<pre>if vXmlNode.AsXmlElement().IsEmpty then</pre>
<pre>vDotNetXMLNode := vDotNetXmlNodeList.Item(0).ChildNodes.Item(0);</pre>	<pre>vXMLNodeList.Get(0, vXmlNode); vXMLNodeList:= vXmlNode.AsXmlElement().GetChildNodes(); vXMLNodeList.Get(0, vXmlNode);</pre>
<pre>Text := vDotNetXMLNode.Attributes.Item(1).InnerText;</pre>	<pre>vXmlNode.AsXmlElement().Attributes.Get(1, vXmlAttribute); Text := vXmlAttribute.AsXmlNode().AsXmlElement().InnerText;</pre>
<pre>vDotNetXMLNode.OwnerDocument</pre>	<pre>vXmlNode.GetDocument</pre>



<code>vDotNetXMLNode := XmlNode.FirstChild;</code>	<code>vXMLNodeList := XMLNode.AsXmlElement().GetChildElements(); vXMLNodeList.Get(0, XMLChildNode);</code>
<code>XMLNode.Attributes.SetNamedItem(XMLNewAttributeNode);</code>	<code>astAttributeIndex := XMLNode.AsXmlElement().Attributes.Count(); XMLNode.AsXmlElement().Attributes.Get(LastAttributeIndex - 1, LocalXmlAttribute); LocalXmlAttribute.AddAfterSelf(XMLNewAttributeNode);</code>
<code>XMLNewAttributeNode := vDotNetXMLNode.OwnerDocument.CreateAttribute(Name);</code>	<code>XMLNode.GetDocument(localXmlDocument); XMLNewAttributeNode := XmlAttribute.Create(Name, Name).AsXmlNode();</code>
<code>vDotNetXMLNode.Prefix;</code>	<code>XMLNode.AsXmlAttribute().NamespacePrefix;</code>
<code>vDotNetXMLNode.Name;</code>	<code>XMLNode.AsXmlAttribute().Name;</code>
<code>vDotNetXMLNode.NamespaceURI;</code>	<code>XMLRootNode.AsXmlElement().NamespaceUri;</code>

```
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;
XmlNode: 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;
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);
```

Encoding: DotNet Encoding;
MessageBytes: DotNet Array;

```
MessageBytes := Encoding.UTF8.GetBytes(TxtToHash);
```

SystemByteArray: Text;

Base64Convert: Codeunit "Base64 Convert";

```
SystemByteArray := Base64Convert.FromBase64(s);
```

MessageBytes: List of [Text];
Convert: Codeunit "Base64 Convert";

```
MessageBytes.Add(Convert.ToBase64(TxtToHash));
```

Http

<pre> HttpRequest: DotNet HttpRequest; HttpResponse: DotNet HttpResponse; StreamReader: DotNet StreamReader; RequestStream: DotNet Stream; HttpRequest := HttpRequest.Create(Url); HttpRequest.Method := 'POST'; HttpRequest.Timeout := 30000; HttpRequest.ContentType := 'application/soap+xml;charset=UTF-8'; XmlDoc.WriteTo(XmlDocText); ByteArray := Encoding.UTF8.GetBytes(XmlDocText); HttpRequest.ContentLength := ByteArray.Length; RequestStream := HttpRequest.GetRequestStream; RequestStream.Write(ByteArray, 0, ByteArray.Length); RequestStream.Close; HttpResponse := HttpRequest.GetResponse; StreamReader := StreamReader.StreamReader(HttpResponse.GetResponseStream); ResponseText := StreamReader.ReadToEnd; </pre>	<pre> ByteArray: Text; HttpRequest: HttpRequestMessage; RequestText: Text; HttpResponse: 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'); HttpRequest.Content(WebContent); HttpRequest.SetRequestUri(Url); HttpRequest.Method := 'POST'; WebClient.Clear(); WebClient.Send(HttpRequest, HttpResponse); WebClient.Timeout(30000); HttpResponse.Content.ReadAs(ResponseText); </pre>
<pre> HttpRequest := HttpRequest.Create(ServiceUrl); HttpRequest.Method := 'POST'; HttpRequest.KeepAlive := true; HttpRequest.AllowAutoRedirect := true; HttpRequest.Headers.Add('SOAPAction', SoapAction); HttpRequest.ContentType := 'text/xml;charset=utf-8'; HttpRequest.Accept := 'text/xml'; HttpRequest.Timeout := 600000; HttpRequest.AutomaticDecompression := DecompressionMethods.GZip; </pre>	<pre> HttpRequest.SetRequestUri(ServiceUrl); HttpRequest.Method('POST'); HttpRequest.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); </pre>
<pre> if not WebRequestHelper.GetWebResponse(HttpRequest, HttpResponse, ResponseInStream, HttpStatusCode, ResponseHeaders, false) then </pre>	<pre> if not Http.Send(HttpRequest, HttpResponse) then </pre>
<pre> HttpResponse.GetResponseStream.CopyTo(ResponseInStream); IStream.ReadText(ResponseText); </pre>	<pre> HttpResponse.Content.ReadAs(ResponseText); </pre>

DotNet_HttpRequest.Credentials := DotNet_CredentialCache;	HttpRequest.GetHeaders(Headers); Headers.Add('Credentials', DotNet_CredentialCache.ToString());
DotNet_HttpRequest.Headers.Add('SOAPAction', SoapAction);	HttpRequest.GetHeaders(Headers); Headers.Add('SOAPAction', SoapAction);
HttpResponse := HttpRequest.GetResponse;	Http.Send(HttpRequest, HttpResponse); HttpResponse.Content.ReadAs(Vehicles);
HttpRequest := HttpRequest.Create(StrSubstNo(Url, Action));	HttpRequest.SetRequestUri(StrSubstNo(Url, Action));

Reimplement codeunit 6224 "XML DOM Management" procedures

XMLElementArray: array[100] of DotNet XmlNode; XMLDOMManagement.SaveXMLDocumentToOutputStream(FileOutputStream, XMLElementArray[1]);	XMLElementArray: array[100] of XmlNode; XMLElementArray[1].GetDocument(XmlDoc); XmlDoc.WriteTo(FileOutputStream);
vDotNetXmlDocument: DotNet XmlDocument; XmlStream: InStream; XMLDOMManagement.LoadXMLDocumentFromInStream(XmlStream, vDotNetXmlDocument);	vXmlNode: XmlNode; XmlStream: InStream; XmlDocument.ReadFrom(XmlStream, vXmlNode); vXmlNode.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)

	<pre>else exit(""); end;</pre>
<pre>XMLDOMManagement.AddDeclaration(vDotNetXMLDocument, '1.0', 'UTF-8', "");</pre>	<pre>Declaration: XmlDeclaration; Declaration := XmlDeclaration.Create('1.0', 'UTF-8', ""); vXMLDocument.AddFirst(Declaration);</pre>

Record TempBlob to codeunit "Temp Blob"

<pre>TempBlobRecord.Init(); TempBlobRecord.Blob := VATReportArchive."Message BLOB";</pre>	<pre>Clear(TempBlobCodeunit); TempBlobCodeunit.FromRecord(VATReportArchive, VATReportArchive.FieldNo("Message BLOB"));</pre>
<pre>TempBlobRecord.CalcFields(Blob); VATReportArchive."Message BLOB" := TempBlobRecord.Blob;</pre>	<pre>var RecRef: RecordRef; RecRef.GetTable(VATReportArchive); TempBlobCodeunit.ToRecordRef(RecRef, VATReportArchive.FieldNo("Message BLOB")); RecRef.SetTable(VATReportArchive);</pre>

```
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)  
{  
  
}
```

<pre>TempBlobRecord.Init(); TempBlobRecord.FromBase64String(base64encodeString);</pre>	<pre>var Base64Convert: Codeunit "Base64 Convert"; OStream: OutStream; TempBlobCodeunit.CreateOutStream(OStream); OStream.WriteText(Base64Convert.FromBase64(base64encodeString));</pre>
<pre>TempBlobRecord.Init(); TempBlobRecord.Blob := "Portal Files"."The File"; TempBlobRecord.CalcFields(Blob); base64encodeString := TempBlobRecord.ToBase64String();</pre>	<pre>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);</pre>

Other breaking changes

Object

Table 37
"Sales Line"

Table 77

"Report
Selections"

Table 77

"Report
Selections"

Obsolete procedure

CrossReferenceNoLookup

FindPrintUsage

FindPrintUsageVendor

Workaround

```
ItemReferenceMgt.SalesReferenceNoLookup (Rec  
);
```

FindReportUsageForCust

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 FindEmailBodyUsageVendor FindEmailBodyUsageForVend
"



Selections

"

Table 77

"Report
Selections PrintWithCheck

PrintWithCheckForCust

"

Table 77

"Report
Selections PrintWithGUIYesNoWithCheck

PrintWithDialogWithCheckForCust

"

Table 77

"Report
Selections PrintWithGUIYesNoWithCheckVend
 or

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		Use table "Item Reference"



Cross Reference” Table 9650 “Custom Report Layout” Page 870	Discontinued
“Social Listening Setup” Page 871	Discontinued
“Social Listening Search Topic” Page 875	Discontinued
“Social Listening FactBox” Page 876	Discontinued
“Social Listening Setup FactBox” Page 5721	Discontinued
“Item Cross	Use Page “Item Reference Entries”



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. SetCustomerNoVisibility

Discontinued, but can be copied to page extension or reimplement to set visibility

Bill-to

FactBox"

Page 9082

"Customer

Statistics SetCustomerNoVisibility

Discontinued, but can be copied to page extension or reimplement to set visibility

FactBox"

Page 9084

"Customer

Details SetCustomerNoVisibility

Discontinued, but can be copied to page extension or reimplement to set visibility

FactBox"

Codeunit

871

"Social

Listening

Management

Discontinued

"



Codeunit
6620
"Copy
Document
Mgt."

CopyPurchLine

CopyPurchDocLine

Codeunit
6620
"Copy
Document
Mgt."

CopySalesLine

CopySalesDocLine

Codeunit
6620
"Copy
Document
Mgt."

PurchHeaderDocType

GetPurchaseDocumentType

Codeunit
6620
"Copy
Document
Mgt."

ArchPurchHeaderDocType

GetPurchaseDocumentType

Codeunit
6620
"Copy
Document
Mgt."

SalesHeaderDocType

GetSalesDocumentType

Codeunit
6620
"Copy

ArchSalesHeaderDocType

GetSalesDocumentType



Document
Mgt.”

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

Codeunit 1 trigger	New codeunit ID	New codeunit name	New method
MakeDateTimeText	41	TextManagement	MakeDateTimeText
GetSeparateDateTime	41	TextManagement	GetSeparateDateTime
MakeDateText	41	TextManagement	MakeDateText
MakeTimeText	41	TextManagement	MakeTimeText
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
ValidateApplicationLanguage	43	LanguageManagement	ValidateApplicationLanguage
LookupApplicationLanguage	43	LanguageManagement	LookupApplicationLanguage
GetGlobalTableTriggerMask	49	GetGlobalTableTriggerMask	
OnGlobalInsert	49	GlobalTriggerManagement	OnGlobalInsert

Codeunit 1 trigger	New codeunit ID	New codeunit name	New method
OnGlobalModify	49	GlobalTriggerManagement	OnGlobalModify
OnGlobalDelete	49	GlobalTriggerManagement	OnGlobalDelete
OnGlobalRename	49	GlobalTriggerManagement	OnGlobalRename
GetDatabaseTableTriggerSetup	49	GlobalTriggerManagement	GetDatabaseTableTriggerSetup
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

Codeunit 1 trigger	New codeunit ID	New codeunit name	New method
OnAfterGetApplicationVersion	9015	Application System Constants	OnAfterGetApplicationVersion
OnRoleCenterOpen	9170	Conf./Personalization Mgt.	OnRoleCenterOpen
OnAfterGetSystemIndicator	79	OnAfterGetSystemIndicator	
OnAfterFindPrinter	44	ReportManagement	OnAfterGetPrinterName
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



Codeunit 1 trigger	New codeunit ID	New codeunit name	New method
OnAfterOnGlobalRename	49	GlobalTriggerManagement	OnAfterOnGlobalRename
OnAfterGetDatabaseTableTriggerSetup	49	GlobalTriggerManagement	OnAfterGetDatabaseTableTriggerSetup
OnAfterOnDatabaseInsert	49	GlobalTriggerManagement	OnAfterOnDatabaseInsert
OnAfterOnDatabaseModify	49	GlobalTriggerManagement	OnAfterOnDatabaseModify
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	



Codeunit 1 trigger	New codeunit ID	New codeunit name	New method
OnValidateUpgradePerDatabase	9900	OnValidateUpgradePerDatabase	
OnValidateUpgradePerCompany	9900	OnValidateUpgradePerCompany	