

Visma Integration för
Visma Administration 500
Visma Administration 1000
Visma Administration 2000
Visma Förening
Referensmanual

Version: 2021.2
2021-09-27
© 2001 - 2021 Visma Spcs AB

AFFÄRSLOGIK	8
INSTALLATION	8
UTVECKLARPAKET	9
DYNAMISK LADDNING	9
LÄNKNING VID KOMPILERINGSTILLFÄLLET	9
ÖVRIGA PROGRAMMERINGSSPRÅK.....	10
REGISTERINFORMATION.....	10
INSTALLATION AV INTEGRATIONSKLIENTER	10
SYNPUNKTER/ÖNSKEMÅL	11
ADK_DATA	11
INLEDNING TILL ADK_DATA.....	11
<i>Fält</i>	12
<i>metaData</i>	13
ADKCREATEDATA.....	15
C++	15
C#.....	15
<i>Information</i>	15
TYPEN EDATA.....	16
ADKADD.....	17
C++	17
C#.....	17
<i>Information</i>	17
DOKUMENTNUMMER I VISSA REGISTER VID SKAPANDE AV DOKUMENT	18
KOPPLA DOKUMENT I KEDJAN OFFERT, ORDER OCH FAKTURA SAMT AVTAL OCH FAKTURA.....	19
<i>Skapa faktura från order</i>	19
<i>Skapa order från offert</i>	20
<i>Skapa faktura från avtal</i>	20
KOPPLA DOKUMENT I KEDJAN BESTÄLLNING, INKOMMANDE FÖLJESDEL OCH LEVERANTÖRSFAKTURA.....	21
<i>Plocka upp rader från beställning på inkommande följesedel</i>	21
<i>Plocka upp rader från beställning och/eller inkommande följesedel på leverantörsfaktura</i>	22
ADKADDEX.....	23
C++	23
C#.....	23
<i>Information</i>	23
ADKSETSORTORDER	23
C++	23
C#.....	23
<i>Information</i>	23
ADKFIND2, ADKFINDEX, ADKFIND	24
C++	24
C#.....	24
<i>Information</i>	24
ADKFIRST	32
C++	32
C#.....	32
<i>Information</i>	32
ADKNEXT	33
C++	33
C#.....	33
<i>Information</i>	33
ADKPREVIOUS	33
C++	33
C#.....	33
<i>Information</i>	34
ADKLAST.....	34
C++	34

C#.....	34
Information.....	34
ADKNEXTEx, ADKPREVIOUSEx, ADKFIRSTEx, ADKLASTEx.....	34
C++	34
C#.....	35
Information.....	35
ADKUPDATE	35
C++	35
C#.....	35
Information.....	35
ADKUPDATEEx	38
C++	38
C#.....	39
Information.....	39
ADKDELETE/ADKDELETERECORD.....	39
C++	39
C#.....	39
Information.....	39
ADKDELETEROW.....	42
C++	42
C#.....	42
Information.....	42
HJÄLPFUNKTIONER SOM EJ UTFÖR DATABASOPERATIONER	43
ADKOPEN	43
C++	43
Information.....	43
Miljövariabler.....	44
Lagra integrationsanvändare i programmet.....	44
Inloggningsdialog i Apiet.....	44
ADKOPEN2	45
C++	45
C#.....	45
Information.....	45
ADKOPENEx	45
C++	45
Information.....	45
ADKOPENEx2	46
C++	46
C#.....	46
Information.....	46
ADKISLOGINACTIVATED.....	46
C++	46
C#.....	46
Information.....	46
ADKCLOSE	47
C++	47
C#.....	47
Information.....	47
ADKCREATEDATA.....	47
C++	47
C#.....	47
Information.....	47
ADKCREATEDATAROW.....	47
C++	48
C#.....	48
Information.....	48
ADKDELETESTRUCT.....	48
C++	48
C#.....	48

<i>Information</i>	48
ADKRESETSTRUCT	49
C++	49
C#.....	49
<i>Information</i>	49
ADK_ERROR.....	49
<i>Deklaration</i>	49
<i>Information</i>	49
ADKGETERRORTEXT.....	50
C++	50
C#.....	50
<i>Information</i>	50
ADKSETFILTER	51
C++	51
C#.....	51
<i>Information</i>	51
<i>Filtersyntax</i>	51
ADKISFILTERABLE	52
C++	52
C#.....	53
<i>Information</i>	53
ADKCLEARFILTERS	53
C++	53
C#.....	53
<i>Information</i>	53
ADKLONGTODATE OCH ADKDATETO LONG	54
<i>Information</i>	54
ADKLONGTODATE.....	54
C++	54
<i>Information</i>	54
ADKDATETO LONG.....	54
C++	54
<i>Information</i>	55
ADKLONGTODATETIME OCH ADKDATETIMETO LONG.....	55
<i>Information</i>	55
ADKLONGTODATETIME.....	55
C++	55
<i>Information</i>	55
ADKDATETIMETO LONG.....	56
C++	56
<i>Information</i>	56
ADKGET	56
<i>Information</i>	56
ADKGETSTR	56
C++	56
C#.....	56
<i>Information</i>	56
ADKGETBOOL	57
C++	57
C#.....	57
<i>Information</i>	57
ADKGETDATA (ADKGETROWDATA).....	58
C++	58
C#.....	58
<i>Information</i>	58
ADKGETDOUBLE (ADKGETNUMERIC)	58
C++	58
C#.....	58
<i>Information</i>	58
ADKGETDATE	59

C++	59
C#.....	59
Information.....	59
ADKSET	59
Information.....	59
ADKSETSTR	60
C++	60
C#.....	60
Information.....	60
ADKSETBOOL	60
C++	60
C#.....	60
Information.....	61
ADKSETDATA.....	61
C++	61
C#.....	61
Information.....	61
ADKSETDOUBLE	62
C++	62
C#.....	62
Information.....	62
ADKSETDATE (ADKSETTIMESTAMP)	62
C++	62
C#.....	62
Information.....	62
ADKGETTYPE.....	63
C++	63
C#.....	63
Information.....	63
ADKGETLENGTH	63
C++	63
C#.....	63
Information.....	64
ADKGETFIELDNAME	64
C++	64
C#.....	64
Information.....	64
ADKGETDATAROW	65
C++	65
C#.....	65
Information.....	65
ADKGETDECIMALS	65
C++	65
C#.....	65
Information.....	65
ADKGETTIMESTAMPFIELD (ADKGETTIMESTAMPFIELDID)	66
C++	66
C#.....	66
Information.....	66
ADKGETLASTFLDID (ADKGETLASTFIELDID).....	66
C++	66
C#.....	66
Information.....	66
ADKGETNROWSFLDID (ADKGETNROWSFIELDID)	67
C++	67
C#.....	67
Information.....	67
ADKGETROWSFLDID (ADKGETROWSFLDID)	67
C++	67
C#.....	67

Information.....	67
ADKGETROWDBID (ADKGETROWDATAID)	68
C++	68
C#.....	68
Information.....	68
ADKISREADWRITE	68
C++	68
C#.....	69
Information.....	69
ADKSETWARNINGFUNCTION (ADKSETWARNINGDELEGATE)	69
C++	69
C#.....	69
Information.....	69
ADKGETVARIANT	69
C++	69
C#.....	69
Information.....	70
ADKGETCUSTOMERPRICE	71
C++	71
C#.....	71
Information.....	71
ADKGETCUSTOMERPRICE2	72
C++	72
C#.....	72
Information.....	72
ADKGETVATPERCENTONARTICLE	73
C++	73
C#.....	73
Information.....	73
ADKGETSIE	73
C++	73
C#.....	74
Information.....	74
ADKGETBOOKINGYEARID	74
C++	74
C#.....	74
Information.....	74
ADKGETVERSION (ADKGETPROGRAMVERSION)	75
C++	75
C#.....	75
Information.....	75
ADKGETCOMPANYNAME	75
C++	75
C#.....	75
Information.....	75
ADKGETCOMPANYNAMEEX	77
C++	77
C#.....	77
Information.....	77
ADKGETCOMPANYSETTING	78
C++	78
C#.....	78
Information.....	78
ADKGETPROGRAMDATABASEVERSION	79
C++	79
C#.....	79
Information.....	79
FUNKTIONER SOM ÄR ERSATTAS MED NYARE VERSIONER	79
ADKGETADMSize	79

C++	79
Information.....	79
BILAGOR.....	81
KODBESKRIVNING.....	81
<i>AdkOpen</i>	81
<i>AdkSetWarningFunction</i>	81
<i>AdkAdd</i>	82
<i>AdkUpdate</i>	91
<i>Grundläggande koncept för åtkomst av poster i databasen</i>	100
<i>AdkFirst</i>	100
<i>AdkNext</i>	102
<i>AdkLast</i>	104
<i>AdkPrevious</i>	106
<i>AdkFind</i>	108
<i>AdkDelete</i>	111
<i>AdkDeleteRow</i>	111
ARBETA MED TIDSSTÄMPLAR.....	112
ARBETA MED PERIODISERINGAR	113
DYNAMISK MENY.....	115
FELHANTERING.....	117
EXCEPTIONS.....	118
UTSKRIFTSVAL I KUNDER, LEVERANTÖRER OCH TILLHÖRANDE DOKUMENT	118
SAMBANDET BLANKSTEG OCH TOMMA VÄRDEN I STRÄNGFÄLT I DATABASEN	118
UTÖKAD LOGGNING	119
ARBETA MED FILTER.....	120
REGISTERBESKRIVNING.....	122
BESKRIVNING AV FELKODER.....	192
ADKNET4WRAPPER	212
INLEDNING.....	212
INFORMATION	212
NAMESPACE SAMT NY FUNKTION I AdkNetWrapper FR O M VERSION 2021.2	212
NYHETER I AdkNet4Wrapper.DLL I VERSION 2019.1	212
FÖLJANDE FUNKTIONER HAR ÖVERLAGRATS MED SAMMA NAMN OCH DESS PARAMETRAR HAR ANPASSATS FÖR .NET FR O M VERSION 2019.1	212
FUNKTIONER MED AVVIKANDE RETURVÄRDEN MOT HUR DE EXPONERATS I Adk.DLL FR O M VERSION 2019.1	214
FÖLJANDE FUNKTIONER FINNS FROM VERSION 2019.2	214
FÖLJANDE FUNKTIONER ÄR NYA OCH ERSÄTTER GAMLA FUNKTIONER I AdkNet4Wrapper.....	214
FUNKTIONER MED SAMMA DEFINITION SOM TIDIGARE VERSIONER AV AdkNet4Wrapper FR O M VERSION 2019.1.....	215
SÖKVÄG TILL GEMENSAMMA FILER RESPEKTIVE FÖRETAGET I AdkNet4Wrapper FR O M VERSION 2019.1 ..	216
ANVÄNDNING AV AdkNet4Wrapper	216
<i>Komma igång</i>	216
<i>Felhantering</i>	216
<i>Öppna företag</i>	217
<i>Stäng företag</i>	217
<i>Skapa datastruktur</i>	217
<i>Hämta/läsa värden från datastruktur</i>	217
<i>Sätta/skriva värden till datastruktur</i>	218
<i>Lägg till data (post)</i>	219
DATATYPER	219
FELKODER	219
ADKSETWARNINGDELEGATE.....	220
ADKNETWRAPPER	221
INLEDNING.....	221
INFORMATION	221
ANVÄNDNING AV AdkNetWrapper	221

<i>Komma igång</i>	221
<i>Felhantering</i>	221
<i>Öppna företag</i>	222
<i>Stäng företag</i>	222
<i>Skapa datastruktur</i>	222
<i>Hämta/läsa värden från datastruktur</i>	222
<i>Sätta/skriva värden till datastruktur</i>	223
<i>Lägg till data (post)</i>	224
DATATYPER	224
FELKODER	225
ADKSetWARNINGDELEGATE	225

Inledning

Produkten Visma Integration finns i fyra varianter:

- **Visma Integration för Visma Administration 500**
- **Visma Integration för Visma Administration 1000**
- **Visma Integration för Visma Administration 2000**
- **Visma Integration för Visma Förening**

I varje produkt ingår dels programmet, t ex Visma Administration 2000, dels ett API i form av en dll-fil.

Vid installation av **Visma Integration för Visma Administration 500/1000/2000** respektive **Visma Integration för Visma Förening** installeras dels själva programvaran, dels API:et. Anledningen till att API:et och programvaran inte kan installeras separat, är för att säkerställa att API:et och aktuellt program är av samma version.

Affärslogik

I **Visma Administration** och **Visma Förening** knyts information samman genom ett antal logiska samband. Detta innebär att data som registreras i en del av programmet automatiskt uppdaterar eller påverkar andra delar av programmet. Om en kundfaktura makuleras påverkas t ex även databastabellen för order och offert. Vidare finns en rad inbyggda kontroller i programmen som motverkar att felaktig data eller orimliga värden registreras.

API:et bygger på de logiska samband som finns i respektive program. Därmed är det inte möjligt att genom API:et importera data om dessa data inte motsvarar de krav som ställs när motsvarande data registreras direkt i programmen. Med andra ord säkerställer API:et en säker datakommunikation.

Visma Integration för Visma Administration och **Visma Integration för Visma Förening** är ett API som jobbar mot alla dessa program. Här finns tillgång till flertalet av programmets tabeller och fält. Om du i ett av programmen försöker påverka fält eller tabeller som inte ingår i det programmet, returneras särskilda felmeddelanden.

Det är t ex inte möjligt att arbeta med offert i **Visma Administration 1000** och därför kommer du få ett felmeddelande om du genom API:et försöker skapa en offert i **Visma Administration 1000**. På samma sätt får du ett felmeddelande om du genom API:et försöker ange ett sorteringsbegrepp för en artikel i **Visma Förening**.

Visma Integration för Visma Administration och **Visma Integration för Visma Förening** är tänkt att användas i enkeltrådade program. Flertrådsprogrammering sker således på eget bevåg.

I bilagan **Registerbeskrivning** finns en fullständig beskrivning över vilka fält och tabeller som kan påverkas i respektive program.

Installation

API:et utgörs huvudsakligen av en fil som heter Adk.dll, denna fil kommer att läggas i den katalog där **Visma Administration 500/1000/2000** respektive **Visma Förening** installeras. Anledningen till att filen läggs där, är att den är knuten till den version av **Visma Administration 500/1000/2000** respektive **Visma Förening** som finns installerad, när en uppdatering av programmet sker kommer även en uppdatering av Adk.dll att göras. **Adk.dll får inte flyttas, det kan då uppstå versionsproblem vid framtida uppdateringar.**

From version 5.1 finns en installation, integrationsklient, för det fallet att det på en dator bara önskas integration och inte hela programmet. Läs mer om detta i installationsanvisningarna för resp produkt.

Det finns även möjlighet att programmera mot .NET. För mer information om detta hänvisas till avsnitten ADKNETWRAPPER och ADKNET4WRAPPER.

From version 2017.1 läggs sökvägen till programkatalogen för **Visma Administration** resp **Visma Förening** till i miljövariabeln Path för att AdkNetWrapper.dll/AdkNet4Wrapper.dll ska hitta dels till Adk.dll och de filer Adk.dll är beroende av.

Utvecklarpaket

Utvecklarpaket består av Referensmanualen, exempelprojekt och följande filer avsedda för native C++-utveckling:

- Adk.h, C++ headerfil där alla funktioner och strukturer finns deklarerade. För att kunna använda funktionerna i Adk.dll måste Adk.h inkluderas.
- AdkError.h, C++ headerfil där alla felkoder finns definierade. AdkError.h bör inkluderas eftersom de specificerade felkoderna bör användas och inte hårdkodade numret direkt.
- Adk.lib, Ett import library för Adk.dll som kan nyttjas för länkning vid kompileringstillfället. Det finns två olika sätt att anropa funktionerna i adk.dll, dynamisk resp statisk laddning.

Dynamisk laddning

Adk.dll laddas dynamiskt i det externa programmet via LoadLibrary där det i anropet pekas ut var Adk.dll finns installerad. För att kunna göra anrop till de olika funktionerna i API:et måste en funktionspekare initieras till motsvarande API-funktion. Detta gör att det inte är nödvändigt att ladda Adk.dll förrän överföring av information ska ske till eller från **Visma Administration 500/1000/2000** respektive **Visma Förening**.

Genom dynamisk laddning är det möjligt att efter installation av det externa programmet låta användaren peka ut var **Visma Administration 500/1000/2000** respektive **Visma Förening** finns installerat och sedan lagra detta i programmet för att sedan användas vid laddning av Adk.dll.

Sökvägar till **Visma Administration 500/1000/2000** respektive **Visma Förening** kan hittas i Windows register, se avsnittet Registerinformation nedan.

Länkning vid kompileringstillfället

Vid kompilering av det externa programmet länkas Adk.lib (import library) med för att lösa upp funktionsanropen i källkoden. På detta sätt behöver inte funktionspekare nyttjas utan anrop kan direkt göras till funktionerna som de finns deklarerade i Adk.h på vanligt sätt. När det externa programmet startar kommer det direkt att försöka ladda Adk.dll vilket innebär att sökvägen till den katalog där **Visma Administration 500/1000/2000** respektive **Visma Förening** finns installerat måste finnas med i systemvariabeln Path.

Adk.lib innehåller således inte någon kod i sig utan innehåller endast information till länkaren om vilka funktioner som finns exporterade i Adk.dll, etc. Om Adk.dll inte finns med i sökvägen kommer inte det externa programmet att starta utan kommer direkt att avslutas med ett felmeddelande om att Adk.dll inte kunde hittas.

I kodexempel som följer har vi använt oss av statisk länkning till Adk.lib för att få så lättöverskådliga exempel som möjligt.

Sökvägar till **Visma Administration 500/1000/2000** respektive **Visma Förening** kan hittas i Windows register, se avsnittet Registerinformation nedan.

Övriga programmeringsspråk

Vårt api kan användas av andra programmeringsspråk som kan ladda en native windows-dll. Här tillhandahåller vi inga förberedda inkapslingar av adk.dll.

I övrigt bör Adk.h kunna vara till ledning för vilka funktioner som exponeras via API:et och deras uppbyggnad.

I beskrivningen av funktioner nedan kommer deklaration anges för C++. I de fall det är aktuellt kommer motsvarande funktionsdeklaration för C# att finnas, så som den ser ut i AdkNet4Wrapper.dll. Funktioner i C# kan ha andra inparametrar och ibland avvikande funktionsnamn.

Läs mer om förändringar i version 2019.2 av AdkNet4Wrapper i särskilt avsnitt sist i denna referensmanual.

Registerinformation

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\SpcsAdm.exe innehåller sökvägen till programfilen för Visma Administration.

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\SpcsFor.exe innehåller sökvägen till programfilen för Visma Förening.

De nycklar som kan vara av intresse för Visma Integration är:

- AdkDll – innehåller sökvägen till katalogen där adk.dll ligger
- CommonFiles – innehåller sökvägen till mappen Gemensamma filer
- AdmIni – innehåller sökvägen till katalogen som innehåller spcsadm.ini respektive spcsfor.ini
- DefaultCompanyPath – standardsökväg för företagskataloger

Nycklarna heter på ett 64-bitars windows

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\App Paths\SpcsAdm.Exe resp

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\App Paths\SpcsFor.Exe.

Installation av integrationsklienter

Fr o m version 5.1 finns en installation som innehåller enbart integrationen. Denna kan användas på en dator istället för en fullständig klientinstallation. Installationen lägger Adk.dll, AdkNetWrapper.dll och AdkNet4Wrapper.dll i den programkatalog som specificerats. Dessutom skrivs registerinformation till datorn enligt avsnittet ovan.

Installation av integrationsklienter är avsedd att användas på datorer där installation av ett fullständigt program inte är önskvärt. Det går inte att installera den vid sidan om en vanlig installation av programmet.

Installationen finns i katalogen *Gemensamma Filer\Integration\Setup* i programmets standardutdelning på servern. Det är Setup.exe som ska köras.

Visma Administration och Visma Förening har fr o m version 5.1 funktionaliteten att automatiskt sätta igång arbetsplatsinstallationer, vid programstart på arbetsplatsen, om ny version har installerats på servern. Motsvarande funktionalitet saknas för integrationsklienter, så här kan kunderna behöva hjälp att få in sin nya version. Detta kan ju göras manuellt, eller via script eller programkod, tex vid felkod ADKE_APP_OLD Setup.exe kan köras som den är eller i läget för tyst installation.

Vi har stöd för tyst installation om den körs via setup.exe. Nedan visas ett exempel på tyst installation av en integrationsklient.

"\\[servernamn]\\SPCS_Administration\\Gemensamma filer\\Integration\\Setup\\setup.exe" /s /v"
 INSTALLDIR="%programfiles%\\SPCS Administration\\"

(Ersätt [servernamn] mot namnet på serverdatorn.)

För mer information om tysta installationer, kontakta vår support.

Synpunkter/önskemål

Synpunkter och önskemål på **Visma Integration för Visma Administration 500/1000/2000** samt **Visma Integration för Visma Förening** kan skickas till support.api.spcs@visma.com.

ADK_DATA

Inledning till ADK_DATA

I kommunikationen mellan den externa produkten och API:et används en datastruktur som benämns ADK_DATA. Strukturen används i de flesta funktioner som API:et tillhandahåller.

För att kunna överföra eller hämta data från **Visma Administration 500/1000/2000** respektive **Visma Förening** måste en ADK_DATA-struktur skapas. Strukturen skapas genom att funktionen `AdkCreateData()` anropas. Denna funktion förklaras senare. I anropet måste typen av datastruktur anges.

De datastrukturer som finns tillgängliga är:

Programdel	Benämning	Rättighet
Artiklar	ADK_DB_ARTICLE	Läsa, skriva och radera
Kunder	ADK_DB_CUSTOMER	Läsa, skriva och radera
Leverantörer	ADK_DB_SUPPLIER	Läsa, skriva och radera
Projekt (4)	ADK_DB_PROJECT	Läsa, skriva och radera
Fakturor	ADK_DB_INVOICE_HEAD	Läsa, skriva och radera rad
Leverantörsfakturor	ADK_DB_SUPPLIER_INVOICE_HEAD	Läsa, skriva och radera rad
Offerter (1)	ADK_DB_OFFER_HEAD	Läsa, skriva och radera rad
Order (2)	ADK_DB_ORDER_HEAD	Läsa, skriva och radera rad
Inventering (2)	ADK_DB_INVENTORY_ARTICLE	Läsa och skriva
Manuell inleverans (2)	ADK_DB_MANUAL_DELIVERY_IN	Läsa, skriva och radera
Manuell utleverans (2)	ADK_DB_MANUAL_DELIVERY_OUT	Läsa, skriva och radera
Artikelgrupp	ADK_DB_CODE_OF_ARTICLE_GROUP	Läsa, skriva och radera
Artikelkonto	ADK_DB_CODE_OF_ARTICLE_ACCOUNT	Läsa
Betalningsvillkor	ADK_DB_CODE_OF_TERMS_OF_PAYMENT	Läsa, skriva och radera
Distrikt (2)	ADK_DB_CODE_OF_DISTRICT	Läsa, skriva och radera
Enhet	ADK_DB_CODE_OF_UNIT	Läsa, skriva och radera
Företagsuppgifter/ föreningsuppgifter	ADK_DB_PRM	Läsa och skriva
Konto	ADK_DB_ACCOUNT	Läsa, skriva och radera
Kundkategori	ADK_DB_CODE_OF_CUSTOMER_CATEGORY	Läsa, skriva och radera
Leveranssätt	ADK_DB_CODE_OF_WAY_OF_DELIVERY	Läsa, skriva och radera
Leveransvillkor	ADK_DB_CODE_OF_TERMS_OF_DELIVERY	Läsa, skriva och radera
Prislista	ADK_DB_CODE_OF_PRICE_LIST	Läsa, skriva och radera
Rabattavtal	ADK_DB_DISCOUNT_AGREEMENT	Läsa, skriva och radera
Resultatenhet	ADK_DB_CODE_OF_PROFIT CENTRE	Läsa, skriva och radera
Språk (3)	ADK_DB_CODE_OF_LANGUAGE	Läsa, skriva och radera
Säljare (2)	ADK_DB_CODE_OF_SELLER	Läsa, skriva och radera
Valuta (3)	ADK_DB_CODE_OF_CURRENCY	Läsa, skriva och radera
Speditör (3)	ADK_DB_DISPATCHER	Läsa, skriva och radera
Beställningar (1)	ADK_DB_BOOKING_HEAD	Läsa, skriva och radera rad
Rabattavtalsrader	ADK_DB_CODE_OF_CUSTOMER_DISCOUNT_ROW	Läsa, skriva och radera
Paketartiklar (1)	ADK_DB_CODE_OF_ARTICLE_PARCEL	Läsa
Artikelbenämningar	ADK_DB_CODE_OF_ARTICLE_NAME	Läsa, skriva och radera
Försäljningspriser	ADK_DB_PRICE	Läsa, skriva och radera
Inköpspriser (2)	ADK_DB_ARTICLE_PURCHASE PRICE	Läsa, skriva och radera
Betalsätt	ADK_DB_CODE_OF_WAY_OF_PAYMENT	Läsa, skriva och radera
Fri kategori 1 (5)	ADK_DB_FREE CATEGORY 1	Läsa, skriva och radera
Fri kategori 2 (5)	ADK_DB_FREE CATEGORY 2	Läsa, skriva och radera

Fri kategori 3 (5)	ADK DB FREE CATEGORY 3	Läsa, skriva och radera
Fri kategori 4 (5)	ADK DB FREE CATEGORY 4	Läsa, skriva och radera
Fri kategori 5 (5)	ADK DB FREE CATEGORY 5	Läsa, skriva och radera
Fri kategori 6 (5)	ADK DB FREE CATEGORY 6	Läsa, skriva och radera
Fri kategori 7 (5)	ADK DB FREE CATEGORY 7	Läsa, skriva och radera
Fri kategori 8 (5)	ADK DB FREE CATEGORY 8	Läsa, skriva och radera
Fri kategori 9 (5)	ADK DB FREE CATEGORY 9	Läsa, skriva och radera
Fri kategori 10 (5)	ADK DB FREE CATEGORY 10	Läsa, skriva och radera
Medlemmar (5)	ADK DB MEMBER	Läsa, skriva och radera
Inkommande följesedel (1)	ADK DB DELIVERY NOTE HEAD	Läsa, skriva och radera rad
Kolli (2)	ADK DB PACKAGE HEAD	Läsa, skriva och radera
Levavikolli (1)	ADK DB IMP PACKAGE HEAD	Läsa
Avvikande levadress(3)	ADK DB DELIVERY ADDRESS	Läsa, skriva och radera
Er referens, kund	ADK DB CODE OF YOUR REF CUSTOMER	Läsa, skriva och radera
Er referens, leverantör	ADK DB CODE OF YOUR REF SUPPLIER	Läsa, skriva och radera
Landskoder	ADK DB CODE OF COUNTRY CODE	Läsa, skriva och radera
Kundinbetalningar	ADK DB CUSTOMERPAYMENT	Läsa, skriva och radera
Bortskrivningskoder	ADK DB CODE OF ADJUSTMENT CODE	Läsa, skriva och radera
Företagsinställningar2	ADK DB PRM2	Läsa
Utbetalningar	ADK DB SUPPLIERPAYMENT	Läsa
Kontakter (3)	ADK DB CONTACT	Läsa, skriva och radera
Verifikationer	ADK DB VERIFICATION HEAD	Läsa och skriva
Bokföringsår	ADK DB CODE OF BOOKINGYEAR	Läsa
Rabattkoder	ADK DB CODE OF DISCOUNT CODE	Läsa och skriva
Kontakttitlar (3)	ADK DB CODE OF CONTACT TITLES	Läsa, skriva och radera
Kontaktgrupper (3)	ADK DB CODE OF CONTACT GROUPS	Läsa, skriva och radera
Kontaktkopplingar (3)	ADK DB CODE OF CONTACT GROUP CONTACTS	Läsa, skriva och radera
Skattereduktion (3)	ADK DB TAX REDUCTION	Läsa, skriva och radera
Avtal (4)	ADK DB AGREEMENT HEAD	Läsa, skriva och radera rad
Skattereduktion på order (2)	ADK DB TAX REDUCTION ORDER	Läsa, skriva och radera
Skattereduktion på avtal (2)	ADK DB TAX REDUCTION AVTAL	Läsa, skriva och radera
Verifikationsserier	ADK DB VERIFICATION SERIES	Läsa
Behandlingshistorik	ADK DB BOOKKEEPINGHIST	Läsa
Periodiseringar	ADK DB PERIODIC ADJUSTMENT	Läsa, Skriva (6)
Kundartiklar	ADK DB CUSTOMER ARTICLE	Läsa, skriva och radera
Kopplade underlag	ADK DB ATTACHMENT INFO	Läsa
Husarbetstyper (3)	ADK DB TAX REDUCTION TYPES	Läsa
Företagsinställningar3	ADK DB PRM3	Läsa

(1) Tabellen finns endast i Visma Administration 2000.

(2) Tabellen finns endast i Visma Administration 1000/2000.

(3) Tabellen finns endast i Visma Administration 500/1000/2000.

(4) Tabellen finns endast i Visma Administration 1000/2000 och Visma Förening.

(5) Tabellen finns endast i Visma Förening

(6) Tabellen öppen för att skapa poster, ej för redigering av befintliga poster

Fält

En ADK_DATA-struktur består av ett antal fält där varje fält motsvarar ett fält i databasen. Antalet fält varierar beroende på vilken typ av ADK_DATA som används, t ex har en kund fler fält än vad som är fallet för en leverantör. För de delar av **Visma Administration 500/1000/2000** respektive **Visma Förening** som är möjliga att anropa via API:et finns de tillgängliga fälten specificerade i bilagan **Registerbeskrivning**. För varje databas finns även definitioner som indikerar första och sista fältet.

Varje fält har flera egenskaper. Dessa ska inte läsas direkt, utan det finns funktioner för att komma åt dessa egenskaper för angivet fält i datastrukturen.

iType

iType beskriver vilken typ fältet är. När en ADK_DATA-struktur skapas antar ett fält en viss typ. Det är ej möjligt att ändra fältets typ. Det finns ett antal fördefinierade typer och ett fält kan endast anta någon av dessa typer. Det är därmed inte möjligt att själv definiera andra typer av fält.

De olika typerna är:

- eUnused – fältet används ej.

- `eBool` – fältet är av typen boolsk.
- `eChar` – fältet är av typen tecken.
- `eDouble` – fältet är av typen flyttal.
- `eDate` – fältet är av typen datum, vilket representeras med heltal.
- `eData` – fältet är av typen datastruktur, dvs en pekare till en `ADK_DATA`.

Fältets typ kan hämtas med funktionen `AdkGetType()`.

bChanged

`bChanged` är en flagga som indikerar om ett fält blivit ändrat. Flaggan sätts till sann när ett fält tilldelas ett värde med hjälp av `AdkSet`-funktionerna.

iFieldId

Fältets id, `iFieldId`, är ett nummer från noll och uppåt och motsvaras av det id som finns i registerbeskrivningen, se bilaga **Registerbeskrivning**.

iInternalFieldId

`iInternalFieldId` är ett fältid för internt bruk.

iLength

`iLength` är fältets längd, d v s hur många tecken det kan innehålla. Fältets längd hämtas med funktionen `AdkGetLength()`. Har fältet decimaler är längden inklusive decimaler och decimalavskiljare.

iRW

`iRW` indikerar om ett fält är `eReadOnly` eller `eReadWrite`. Värdet `eReadOnly` innebär att det endast går att läsa data från fältet. Värdet `eReadWrite` innebär att det är tillåtet att både läsa och skriva i ett fält. Genom funktionen `AdkIsReadWrite()` är det möjligt att kontrollera om ett fält är `eReadWrite` eller ej.

Value

`Value` är det värde ett fält innehar. `Value` är en union och kan innehålla samtliga de datatyper som beskrivits ovan. Att tilldela ett värde till ett fält sker genom de olika `AdkSet`-funktionerna. Där väljs den funktion som passar ihop med fältets typ. Att läsa ett värde i ett fält sker genom de olika `AdkGet`-funktionerna. Även här väljes den funktion som passar ihop med fältets typ.

metaData

Förutom fälten innehåller `ADK_DATA`-strukturen information som underlättar kommunikationen mellan den externa produkten och API:et, `metaData`. `metaData` innehåller följande information om `ADK_DATA`:

- `hHandle` - internt "handtag" till den session mot databasen `ADK_DATA`-strukturen använder.
- `iDatabaseId` - vilken typ av datastruktur som används.
- `iFields` - det antal fält som datastrukturen innehåller.

- `iStructSize` - datastrukturens storlek.

Den information som `metaData` innehåller används endast för internt bruk av API:et och ska därför inte redigeras.

AdkCreateData

C++

```
PADK_DATA AdkCreateData (INT iDatabaseId)
```

C#

```
int AdkCreateData (int IDataBaseId)
```

Information

För att skapa en ADK_DATA-struktur måste först en pekare deklarerats, en PADK_DATA. Detta sker med hjälp av följande kodrad:

```
PADK_DATA pData;
```

När pekaren är skapad kan den initieras och anta den typ av ADK_DATA som önskas. För initiering finns två olika funktioner `AdkCreateData()` och `AdkCreateDataRow()`. I de flesta fallen används `AdkCreateData()`. `AdkCreateDataRow()` används endast vid skapandet av rader till order, offert, faktura, leverantörsfaktura, inkommande följesedel, beställning, kolli, verifikation samt avtal. För mer information om funktionen `AdkCreateDataRow()`, se avsnittet med samma namn.

Funktionen `AdkCreateData()` tar en inparameter, vilken anger vilken typ av datastruktur som ska skapas. Alla de typer av datastrukturer som kan skapas specificeras i filen `Adk.h`, se tabell 1. Nedan visas hur `pData` initieras när avsikten är att skapa en datastruktur av typen "Kund".

```
pData = AdkCreateData (ADK_DB_CUSTOMER);
```

Efter initieringen pekar `pData` på en datastruktur av typen "Kund" och har nu fått diverse värden. `metaData`, dvs informationen om datastrukturen, har följande innehåll:

- `iDatabaseId = 0`. Kundtabellen har id 0.
- `iFields = 136`. Kund innehåller 136 fält med id 0 - 135.

Det har nu skapats 136 fält med de egenskaper som tidigare beskrivits. Dessa fält har fått id 0 - 135. Datastrukturen `pData` innehåller 200 fält men endast de 136 första av dessa används. Här följer en beskrivning av fält nummer 0, fältet för kundnummer:

- `iType = 2`. Vilket innebär typ nummer 2 som är `eChar`.
- `bChanged = 0`. Oförändrad.
- `iFieldId = 0`. Fältets id är 0.
- `iInternalFieldId = 0`. Fältets internid är 0.
- `iLength = 16`. Fältet kan innehålla en 16 tecken lång sträng.
- `iRW = eReadWrite`. Det är tillåtet att läsa och skriva i fältet.
- `value.s = ""`. Värdet är en tom sträng. Om fältet är av datatypen `double` kommer `value.d` att anta ett värde.

I .NET representeras PADK_DATA av `int`. Kodexempel C#:


```
int pData;
pData = AdkCreateData(ADK_DB_CUSTOMER);
```

Typen eData

Den information som hittills beskrivits om datastrukturen ADK_DATA är fullt tillräcklig i de fall där arbete inte ska ske med offert, order, faktura, leverantörsfaktura, inkommande följesedel, beställning, verifikation, avtal eller kollihantering. Dessa programdelar är annorlunda i det avseendet att de kan bestå av ett dokumenthuvud med ett antal tillhörande dokumentrader. Kopplingen mellan dokumenthuvudet och dokumentraderna sker med hjälp av en pekare. Pekaren finns i dokumenthuvudet och den kopplar samman de två olika delarna av dokumentet.

1. Faktura		2. Kunduppgifter		3. Moms, konton		4. Spårning	
Fakturanr:	891	Faktura	Betvillk:	N	Fakturadatum:	2014-11-26	on
Kundnr:	113		Levvillk:	FVL	Förfallodatum:	2014-12-26	fr
Namn:	Storstadshotellet AB		Levsätt:	P	Leveransdatum:	2014-11-26	on
Er referens:	Karin Nilsson		Speditör:		Ert ordernr:		
Vår referens:	Siw Eriksson		Resenh:		Referenskod:		
Kontraktsnr:			Projekt:				

Artikelnummer	Streckkod	Benämning	Beställt antal	Enhet	Levererat antal	å-pris	Rabatt
102	7330256101019	Redovisning Delta	1	st	1	8 900,00	0,00 %
302		Serviceavtal Delta	1	st	1	2 200,00	0,00 %
311		Utbildning	16,0	tim	16,0	900,00	0,00 %

På bilden ovan finns ett dokumenthuvud som består av ett antal fält, Kundnr, Namn etc. Till huvudet har kopplats tre rader, bestående av artiklarna med nummer 102, 302 och 311.

I dokumenthuvudet för order, offert eller faktura finns två extra fält, ADK_OOI_HEAD_ROWS och ADK_OOI_HEAD_NROWS. Även i dokumenthuvudet för leverantörsfaktura finns två extra fält, ADK_SUP_INV_HEAD_ROWS och ADK_SUP_INV_HEAD_NROWS och i beställning finns ADK_BOOKING_ROWS och ADK_BOOKING_NROWS. I verifikation heter fälten ADK_VERIFICATION_HEAD_ROWS resp ADK_VERIFICATION_HEAD_NROWS, i kolli ADK_PACKAGE_HEAD_ROWS resp ADK_PACKAGE_HEAD_NROWS och i inkommande följesedel ADK_DELIVERY_NOTE_ROWS resp ADK_DELIVERY_NOTE_NROWS. I avtal heter fälten ADK_AGREEMENT_HEAD_ROWS resp ADK_AGREEMENT_HEAD_NROWS.

Fälten med benämningarna ADK_OOI_HEAD_NROWS, ADK_SUP_INV_HEAD_NROWS, ADK_BOOKING_NROWS, ADK_VERIFICATION_HEAD_NROWS resp ADK_PACKAGE_HEAD_NROWS, ADK_DELIVERY_NOTE_NROWS och ADK_AGREEMENT_HEAD_NROWS är av typen eDouble och ska tilldelas det antal rader som ska kopplas till dokumenthuvudet, vilket sker med funktionen AdkSetDouble().

Fälten med benämningarna ADK_OOI_HEAD_ROWS, ADK_SUP_INV_HEAD_ROWS, ADK_BOOKING_ROWS, ADK_PACKAGE_HEAD_ROWS, ADK_VERIFICATION_HEAD_ROWS, ADK_DELIVERY_NOTE_ROWS och ADK_AGREEMENT_HEAD_ROWS är av typen eData, vilka kan anta en pekare till en ADK_DATA. Genom att låta något av de fälten anta en pekare till en datastruktur med rader, skapas en koppling mellan dokumenthuvud och dokumentrader. Kopplingen mellan dokumenthuvud och dokumentrad skapas genom funktionen AdkSetData(), vilken tilldelar en pekare till fälten ADK_OOI_HEAD_ROWS, ADK_SUP_INV_HEAD_ROWS, ADK_BOOKING_ROWS, ADK_PACKAGE_HEAD_ROWS, ADK_VERIFICATION_HEAD_ROWS, ADK_DELIVERY_NOTE_ROWS och ADK_AGREEMENT_HEAD_ROWS.

För att skapa ett antal rader används funktionen `AdkCreateDataRow()`. `AdkCreateDataRow()` fungerar på samma sätt som funktionen `AdkCreateData()`, med den skillnaden att den kräver ytterligare en inparameter. Inparametern är av typen `int` och talar om antalet datastrukturer/dokumentrader som ska skapas.

Det är alltså möjligt att skapa en vektor av rader. Om t ex fem rader ska skapas anges värdet 5 som parameter nummer två till `AdkCreateDataRow()`. `AdkCreateDataRow()` returnerar då en pekare till en datastruktur som är en vektor med de fem raderna.

Den pekare som nu innehåller de fem raderna kopplas samman med dokumenthuvudet genom funktionen `AdkSetData()`. Möjligheten att skapa datastrukturer som är vektorer ska endast användas till hanteringen av rader för order, offert, faktura och leverantörsfaktura. För att peka ut och arbeta med en viss rad finns funktionen `AdkGetDataRow()` som returnerar en pekare till en enskild rad. Ytterligare information om de hjälpfunktioner som tagits upp här presenteras senare.

AdkAdd

C++

```
ADK_ERROR AdkAdd(ADK_DATA* pData)
```

C#

```
ADKERROR AdkAdd(int pDb)
```

Information

Funktionen `AdkAdd()` används för att addera en post till databasen. Funktionen `AdkAdd()` tar en inparameter, en pekare till en `ADK_DATA`-struktur. Det är typen på den inskickade datastrukturen som avgör vad som ska adderas till databasen. Är den av typen "Artikel" kommer en artikel att läggas till osv. Denna funktion är möjlig på följande typer av datastrukturer:

- Offert
- Order
- Kundfaktura (normalfaktura, kontantnota och kreditfaktura)
- Leverantörsfaktura (normalfaktura och kreditfaktura)
- Inkommande följesedel
- Beställning
- Kund
- Medlem
- Artikel
- Leverantör
- Projekt
- Inventering
- Manuell inleverans
- Manuell utleverans
- Säljare
- Distrikt
- Kundkategori
- Betalningsvillkor
- Leveransvillkor
- Leveranssätt
- Fri kategori 1 .. Fri kategori 10
- Kolli
- Avvikande leveransadress

- Er referens, kund
- Er referens, leverantör
- Landskoder
- Kunderbetalningar
- Valuta
- Artikelgrupper
- Enhetskoder
- Resultatenheter
- Inköspriser
- Betalsätt
- Verifikationer
- Kontakter
- Kontaktgrupper
- Kontakttitlar
- Kontaktkopplingar
- Språk
- Skattereduktion (Husarbete på faktura/order/avtal)
- Avtal
- Konto
- Periodisering
- Kundartiklar

Är den datastruktur som skickas till funktionen av annan typ, returneras ett felmeddelande som säger att det har skickats in en felaktig datatyp.

De värden som finns i datastrukturen och som skickas in till `AdkAdd()` kommer att lagras som en post i databasen. I `AdkAdd` utförs ett antal olika valideringar beroende på vilken typ av datastruktur som skickats in och vilka fält som antagit värden. Varje fält i en datastruktur är antingen av typen `ReadWrite` eller `ReadOnly`. I fält av typen `ReadWrite` är det tillåtet att tilldela värden. Är fältet däremot av typen `ReadOnly` är det inte tillåtet att tilldela ett värde. `AdkAdd()` returnerar ett felmeddelande om ett värde har tilldelats ett fält som är `ReadOnly`.

Fälten valideras och skrivs till databasen i stigande ordning efter id-nummer, d v s fältet med id = 0 kommer att valideras och skrivas först. Om t ex en faktura skapas och flaggan för utskriven sätts till sann kommer det ej bli möjligt att sätta fler värden på fakturan. Om värden tilldelas efter att flaggan för utskriven har satts till sann kommer ett felmeddelande att returneras och inget kommer att skrivas till databasen. I **Visma Administration 500/1000/2000** respektive **Visma Förening** är det möjligt att ange ett antal standardvärden till olika fält och dessa antas då vid skapande av en ny post. Om ett sådant fält ej har tilldelats ett värde kommer det att anta det standardvärde som finns inlagt i **Visma Administration 500/1000/2000** respektive **Visma Förening**. Funktionen utförs i formen av en transaktion, d v s om fel uppstår returneras ett felmeddelande och inget skrivs till databasen. `AdkAdd()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

Dokumentnummer i vissa register vid skapande av dokument

Vissa register har dokumentnummer som sätts med hjälp av räknare i programmet. Huvudregeln för dessa är att fältet för dokumentnummer ska lämnas blankt när ett dokument skapas via `AdkAdd`.

Det är möjligt om så önskas att sätta ett eget nummer, men då bör dels kontroll göras att detta nummer inte redan finns använt för dokumenttypen i fråga, och dels att det inte lägger beslag på dokumentnummer som programmets interna räknare kommer att använda.

Nästa nummer för de olika dokumenttyperna återfinns i programmet under Arkiv – Företagsunderhåll – Företagsinställningar. Välj noden Nummerserier.

Register detta berör och aktuellt fält:

- Offert (ADK_OOI_HEAD_DOCUMENT_NUMBER)
- Order (ADK_OOI_HEAD_DOCUMENT_NUMBER)
- Kundfaktura (ADK_OOI_HEAD_DOCUMENT_NUMBER)
- Leverantörsfaktura (ADK_SUP_INV_HEAD_GIVEN_NUMBER)
- Inkommande följesedel (ADK_DELIVERY_NOTE_DOCUMENT_NUMBER)
- Beställning (ADK_BOOKING_DOCUMENT_NUMBER)
- Avtal (ADK_AGREEMENT_HEAD_DOCUMENT_NUMBER)
- Manuell inleverans (ADK_MANUAL_DELIVERY_IN_DOCUMENT_NUMBER)
- Manuell utleverans (ADK_MANUAL_DELIVERY_OUT_DOCUMENT_NUMBER)

Koppla dokument i kedjan offert, order och faktura samt avtal och faktura

Skapa faktura från order

Att skapa en faktura från en order görs med `AdkAdd()` på en `AKD_DATA`-struktur för fakturatabellen.

När en faktura skapas i **Visma Integration för Visma Administration 1000/2000** går det att välja om en faktura ska skapas från en order genom att ange ett giltigt dokumentnummer i fältet `ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER1`. Då skapas fakturan på samma sätt som om kommandot Skapa faktura väljs i orderfönstret i programmet.

Det som kan skapas är normala fakturor (fakturatyp 'F') och kontantnotor (fakturatyp 'N'). Kreditfakturor kan inte skapas från order.

Vid skapandet av fakturan tas raderna med från ordern till fakturan. Samma kontroller som i programmet görs, t ex beräknas hur många artiklar som kan levereras på fakturaraden med ledning av antal i lager för artikeln. Antal fakturarader som önskas för fakturan kan inte anges, utan detta sköts av API:et. Det är dock möjligt att senare ändra i fakturan om t ex fler rader ska läggas till.

När en faktura skapas från order är det möjligt att ange ett eget fakturanummer på samma sätt som när en faktura skapas fristående.

Transaktionen avbryts och ingen faktura skapas om:

- fakturatyp inte är 'F' eller 'N'
- ordern inte existerar
- ordern är makulerad
- ordern redan är fakturerad
- angivet antal rader (`ADK_OOI_HEAD_NROWS`) för fakturan inte är noll
- värde har angivits i fältet `ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER2`

När faktura skapas från order i **Visma Integration för Visma Administration 2000** kan restorder skapas automatiskt. Detta görs om ordern ifråga har restnoterade rader och flaggan för restnotera ej (`ADK_OOI_HEAD_DO_NOT_CREATE_BACKORDER`) inte är sann. I programmet kan detta påverkas via kommandot skapa order, men i API:et finns inte denna möjlighet. Möjligheten finns dock att ändra i ordern via `AdkUpdate` innan fakturan skapas, så att fältet restnotera ej är sant, vilket leder till att ingen restorder skapas även om det skulle finnas restnoterade rader.

Funktionalitet från tidigare versioner av API:et där det gick att ändra kopplingsdokuments-fälten via `AdkUpdate` har tagits bort från API:et. Om `ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER1` eller `ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER2` är ändrat avbryts transaktionen och ett felmeddelande genereras.

Fält som kan användas när faktura skapas från order är:

- `ADK_OOI_HEAD_DOCUMENT_NUMBER`

- ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER1
- ADK_OOI_HEAD_TYPE_OF_INVOICE

Fält som inte får ändras:

- ADK_OOI_HEAD_NROWS
- ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER2

Observera att att det inte går att skapa en faktura från en order och ett avtal på samma dokument.

Övriga fält av typen `ReadWrite` i huvudet på fakturan ignoreras helt när faktura skapas på detta sätt. Inga fel eller varningar genereras.

Skapa order från offert

Att skapa en order från en offert görs med `AdkAdd()` på en `AKD_DATA`-struktur för ordertabellen.

När en order skapas i **Visma Integration för Visma Administration 2000** går det att styra om ordern ska skapas från en offert genom att ange ett giltigt dokumentnummer i fältet `ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER1`. Då skapas order på samma sätt som om kommandot Skapa order väljs i offertfönstret i programmet.

Vid skapandet av ordern tas raderna med från offerten till ordern. Antal orderrader som önskas för ordern kan inte anges, utan detta sköts av API:et. Det är dock möjligt att senare ändra i ordern om t ex fler rader ska läggas till.

När en order skapas från offert är det möjligt att ange ett eget ordernummer på samma sätt som när en order skapas fristående.

Transaktionen avbryts och ingen order skapas om:

- offerten inte existerar
- offerten är makulerad
- offerten redan ligger till grund för en order
- angivet antal rader för ordern (`ADK_OOI_HEAD_NROWS`) inte är noll

Funktionalitet från tidigare versioner av API:et där det gick att ändra kopplingsdokuments-fälten via `AdkUpdate` har tagits bort från API:et. Försök att ändra `ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER1` eller `ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER2` leder till att transaktionen avbryts och ett felmeddelande genereras.

Fält som kan användas när order skapas från offert är:

- ADK_OOI_HEAD_DOCUMENT_NUMBER
- ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER1

Fält som inte får ändras:

- ADK_OOI_HEAD_NROWS

Övriga fält av typen `ReadWrite` i huvudet på ordern ignoreras helt när order skapas på detta sätt. Inga fel eller varningar genereras.

Skapa faktura från avtal

Att skapa en faktura från ett avtal görs med `AdkAdd()` på en `AKD_DATA`-struktur för fakturatabellen.

När en faktura skapas i **Visma Integration för Visma Administration 1000/2000** och **Visma Integration för Visma Förening** går det att välja om en faktura ska skapas från ett avtal genom att ange ett giltigt dokumentnummer i fältet `ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER2`.

Vid skapandet av fakturan tas raderna med från avtalet till fakturan. Samma kontroller som i programmet görs, t ex beräknas priser om så är inställt på avtalet. Antal fakturarader som önskas för fakturan kan inte anges, utan detta sköts av API:et. Det är dock möjligt att senare ändra i fakturan om t ex fler rader ska läggas till.

När en faktura skapas från avtal är det möjligt att ange ett eget fakturanummer på samma sätt som när en faktura skapas fristående.

Transaktionen avbryts och ingen faktura skapas om:

- avtalet inte existerar
- avtalet är makulerat
- avtalet är avslutat
- avtalet är markerat som Ej klar
- angivet antal rader (`ADK_OOI_HEAD_NROWS`) för fakturan inte är noll
- värde har angivits i fältet `ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER1`

Värde måste också anges i `ADK_OOI_HEAD_AGREEMENT_DATE_TYPE` och ska vara ett av följande värden:

- "G" – Tilldelat datum i `ADK_OOI_HEAD_DOCUMENT_DATE1`
- "T" – Dagens datum
- "N" – Nästa fakturadatum på avtalet

Fält som kan användas när faktura skapas från avtal är:

- `ADK_OOI_HEAD_DOCUMENT_NUMBER`
- `ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER2`
- `ADK_OOI_HEAD_AGREEMENT_DATE_TYPE`
- `ADK_OOI_HEAD_DOCUMENT_DATE1`

Fält som inte får ändras:

- `ADK_OOI_HEAD_NROWS`
- `ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER1`

Observera att det inte går att skapa skapa en faktura från en order och ett avtal på samma dokument.

Övriga fält av typen `ReadWrite` i huvudet på fakturan ignoreras helt när faktura skapas på detta sätt. Inga fel eller varningar genereras.

Koppla dokument i kedjan beställning, inkommande följesedel och leverantörsfaktura

Plocka upp rader från beställning på inkommande följesedel

Att plocka upp rader från en beställning görs med `AdkAdd()` eller `AdkUpdate()` på en `AKD_DATA`-struktur för inkommande följesedelstabellen.

När en rad från en beställning ska plockas upp på en inkommande följesedel skapad i **Visma Integration för Visma Administration 2000** ska ett giltigt dokumentnummer anges i fältet `ADK_OOI_ROW_FROM_DOCUMENT` samt vilken typ av dokument, beställning i detta fall, anges i fältet `ADK_OOI_ROW_FROM_TYPE`. Finns dessa med på en rad läggs beställningen upp bland de beställningar som följesedeln hämtat raderna från. Dessa beställningar ser du i **Visma Administration 2000** under Arbets med – Artiklar/Lager – Inkommande följesedel, fliken 2. Beställningar, översta tabellen.

När följesedel skapas eller uppdateras kan raderna tas med från beställningen genom att fältet `ADK_OOI_ROW_FROM_DOCROW` tilldelas, vilket även medför att samma kontroller som i programmet görs. Det är även möjligt att senare ändra på följesedeln om t ex fler rader ska läggas till.

När en inkommande följesedel skapas och rader samtidigt plockas upp från beställning är det möjligt att ange ett eget löpnummer på samma sätt som när en följesedel skapas fristående.

Transaktionen avbryts och följesedel skapas eller uppdateras inte om:

- dokumenttypen inte är 'B' som i beställning
- beställning inte existerar
- beställning är makulerad
- beställning redan är uppläst
- angiven beställningsrad inte existerar

Funktionaliteten att via API:et kunna ändra kopplingsdokuments-fälten via `AdkUpdate` är inte tillgänglig.

Fält som kan användas när rader plockas upp från beställning på inkommande följesedel är:

- `ADK_OOI_ROW_FROM_DOCUMENT`
- `ADK_OOI_ROW_FROM_TYPE`
- `ADK_OOI_ROW_FROM_DOCROW`

Plocka upp rader från beställning och/eller inkommande följesedel på leverantörsfaktura

Att plocka upp rader från en beställning och/eller inkommande följesedel görs med `AdkAdd()` eller `AdkUpdate()` på en `AKD_DATA`-struktur för leverantörsfakturatabellen.

När en rad från en beställning eller en inkommande följesedel ska plockas upp på en leverantörsfaktura i **Visma Integration för Visma Administration 2000** ska ett giltigt dokumentnummer anges i fältet `ADK_OOI_ROW_FROM_DOCUMENT` samt typ av dokument anges i fältet `ADK_OOI_ROW_FROM_TYPE`.

När leverantörsfaktura skapas eller uppdateras kan raderna tas med från beställningen och/eller inkommande följesedeln genom att fältet `ADK_OOI_ROW_FROM_DOCROW` tilldelas, vilket även medför att samma kontroller som i programmet görs. Det är även möjligt att senare ändra på leverantörsfakturan om t ex fler rader ska läggas till.

När en leverantörsfaktura skapas och rader samtidigt plockas upp från beställning eller en följesedel är det möjligt att ange ett eget löpnummer på samma sätt som när en leverantörsfaktura skapas fristående.

Transaktionen avbryts och ingen leverantörsfaktura skapas om:

- dokumenttypen inte är 'B' eller 'I'
- beställning/inkommande följesedel inte existerar
- beställning/inkommande följesedel är makulerad
- beställning/inkommande följesedel redan är uppläst
- angiven beställningsrad/följesedelsrad inte existerar

Funktionaliteten att via API:et kunna ändra kopplingsdokuments-fälten via `AdkUpdate` är inte tillgänglig.

Fält som kan användas när rader plockas upp från beställning/följesedel på leverantörsfaktura är:

- `ADK_OOI_ROW_FROM_DOCUMENT`
- `ADK_OOI_ROW_FROM_TYPE`
- `ADK_OOI_ROW_FROM_DOCROW`

AdkAddEx

C++

```
ADK_ERROR AdkAddEx(ADK_DATA* pData, BOOL bRow)
```

C#

```
ADKERROR AdkAddEx(int pDb, bool bRow)
```

Information

Fungerar exakt som sin motsvarighet med tillägget att det går att välja om rader ska läsas in eller inte om sådana finns (ex på fakturor) efter att operationen ifråga utförts. Detta är gjort för att det ska gå snabbare att lägga till poster i tabeller med kopplade rader. Effekten blir tydligare när databasanrop går lite långsammare, tex över nätverk. Det är den andra parametern i anropet som bestämmer huruvida rader ska tas med eller inte. Värdet `TRUE` anger att raderna ska tas med i svaret. Värdet `FALSE` anger att raderna inte ska tas med i svaret. (`TRUE` = 1, `FALSE` = 0 enligt definition i windows headerfiler). Om rader inte tas med hämtas således endast huvudet.

AdkAddEx() finns i *AdkNet4Wrapper* i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är *AdkNet4Wrapper.dll* bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkSetSortOrder

C++

```
ADK_ERROR AdkSetSortOrder(ADK_DATA* pData, ADK_SORT_ORDER nSortOrder)
```

C#

```
ADKERROR AdkSetSortOrder(int pDb, ADK_SORT_ORDER nSortOrder)
```

Information

Funktionen `AdkSetSortOrder()` bestämmer vilken sorteringsordning som ska användas till den databastabell som aktuell datastruktur använder. `AdkSetSortOrder()` tar två inparametrar. Den första inparametern är en pekare till en datastruktur, ur vilken funktionen tar fram den tabell där sorteringsordningen ska ändras. Det som är väsentligt i denna datastruktur är dess typ, om den innehåller värden eller ej saknar betydelse. Den andra inparametern anger den sorteringsordning som önskas. Denna sorteringsordning måste passa till den tabell som är kopplad till datastrukturen. Om så ej är fallet kommer ett felmeddelande att returneras.

De sorteringsordningar som finns tillgängliga och vilka typer av datastrukturer dessa passar ihop med, finns specificerat nedan samt i `Adk.h`-filen. Denna funktion är möjlig att utföra på alla deklarerade typer av datastrukturer. `AdkSetSortOrder()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkFind2, AdkFindEx, AdkFind

AdkFindEx och **AdkFind** är kvar för bakåtkompatibilitet. Vi rekommenderar att **AdkFind2** används istället.

C++

```
ADK_ERROR AdkFind2(ADK_DATA* pData, BOOL bRow)
ADK_ERROR AdkFindEx(ADK_DATA* pData, BOOL bRow)
ADK_ERROR AdkFind(ADK_DATA* pData)
```

C#

```
ADKERROR AdkFind2(int pDb, bool bRow)
ADKERROR AdkFindEx(int pDb, bool bRow)
ADKERROR AdkFind(int pDb)
```

Information

Funktionen `AdkFind()` söker information i någon av de programdelar som anges i tabell 1. `AdkFind()` tar en inparameter, en pekare till en datastruktur. Denna datastruktur fylls med data från den första posten som motsvarar angivet sökvärde.

För att söka måste först önskad sökordning bestämmas. Detta görs med hjälp av funktionen `AdkSetSortOrder()` som beskrivs ovan. Om inte någon sorteringsordning anges är det den sorteringsordning som är default som gäller. Sorteringsordningen för att söka bland kunder kan t ex anges till kundnummer eller kundnamn. Därefter kan funktionen `AdkFind()` användas för att söka i den definierade sorteringsordningen. Sökvärdet tilldelas det fält som ska kopplas samman med aktuell sökordning. Den inskickade datastrukturen kommer att kompletteras med övriga värden som den funna posten innehåller.

Funktionen `AdkFindEx()` fungerar exakt som sina motsvarigheter med tillägget att det går att välja om rader ska läsas in eller inte om sådana finns (ex på fakturor). Detta är gjort för att det ska gå snabbare att söka igenom tabeller med kopplade rader. Det är den andra parametern i anropet som bestämmer huruvida rader ska tas med eller inte. Värdet `TRUE` anger att raderna ska tas med i svaret. Värdet `FALSE` anger att raderna inte ska tas med i svaret. (`TRUE` = 1, `FALSE` = 0 enligt definition i windows headerfiler). Om rader inte tas med hämtas således endast huvudet.

Funktionen `AdkFind2()` fungerar som `AdkFindEx()` med två separata förbättringar.

AdkFind2(), AdkFindEx() finns i AdkNet4Wrapper i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

För ett antal tabeller kan sökning göras mer träffsäker då en jämförelse görs att den träffade posten har exakt samma värde i sökfältet som sökvärdet. Detta har att göra med att `AdkFind` ger träff på inledande värde vid sökning mot strängfält.

ADK_DB_ARTICLE
ADK_DB_CUSTOMER
ADK_DB_SUPPLIER
ADK_DB_CODE_OF_TERMS_OF_PAYMENT
ADK_DB_CODE_OF_LANGUAGE
ADK_DB_CODE_OF_CURRENCY
ADK_DB_CODE_OF_CUSTOMER_CATEGORY
ADK_DB_CODE_OF_DISTRICT

ADK_DB_CODE_OF_SELLER
ADK_DB_DISCOUNT_AGREEMENT
ADK_DB_CODE_OF_UNIT
ADK_DB_CODE_OF_PROFIT_CENTRE
ADK_DB_CODE_OF_PRICE_LIST
ADK_DB_DISPATCHER
ADK_DB_MEMBER
ADK_DB_CODE_OF_COUNTRY_CODE
ADK_DB_CODE_OF_TERMS_OF_DELIVERY
ADK_DB_CODE_OF_WAY_OF_DELIVERY
ADK_DB_CODE_OF_CONTACT_TITLES
ADK_DB_CODE_OF_CONTACT_GROUPS

För andra tabeller går det att till vissa sökordningar att ange sökvärden i fler fält som sökbegrepp för en mer exakt träff i databasen.(Vissa av dessa har fungerat i AdkFind sedan tidigare).

Tabell	Sökordning	Fält
ADK_DB_ARTICLE_PURCHASE_PRICE	ePurchasePriceArticleNrBase, ePurchasePriceSupplierNrBase	ADK_ARTICLE_PURCHASE_PRICE_ARTICLE_NUMBER,ADK_ARTICLE_PURCHASE_PRICE_SUPPLIER_NUMBER
	ePurchasePriceArticleNrStf, ePurchasePriceSupplierNrStf	ADK_ARTICLE_PURCHASE_PRICE_ARTICLE_NUMBER,ADK_ARTICLE_PURCHASE_PRICE_SUPPLIER_NUMBER, ADK_ARTICLE_PURCHASE_PRICE_QUANTITY
ADK_DB_CODE_OF_ARTICLE_NAME	eArticleNameNr	ADK_CODE_OF_ARTICLE_NAME_ARTICLE_NUMBER, ADK_CODE_OF_ARTICLE_NAME_LANGUAGE_CODE
ADK_DB_CODE_OF_DISCOUNT_CODE	eCodeOfDiscount	ADK_CODE_OF_DISCOUNT_FILTER, ADK_CODE_OF_DISCOUNT_CODE
ADK_DB_CODE_OF_CONTACT_GROUP_CONTACTS	eCodeOfContactsAndContactsGroup, eCodeOfContactsAndContactsGroupGroupID	ADK_CODE_OF_CONTACT_GROUP_CONTACTS_CONTACT_ID, ADK_CODE_OF_CONTACT_GROUP_CONTACTS_GROUP_ID
ADK_DB_CONTACT	eContactCustomerNr	ADK_CONTACTS_CUSTOMER_NUMBER, ADK_CONTACTS_NAME
	eContactSupplierNr	ADK_CONTACTS_SUPPLIER_NUMBER, ADK_CONTACTS_NAME
	eContactCustomerSign	ADK_CONTACTS_CUSTOMER_NUMBER, ADK_CONTACTS_SIGN
	eContactSupplierSign	ADK_CONTACTS_SUPPLIER_NUMBER, ADK_CONTACTS_SIGN
ADK_DB_PRICE	ePriceArticleNr, ePriceBaseArticleNr, ePriceNotBaseArticleNr, ePriceList, ePriceBaseList, ePriceNotBaseList	ADK_PRICE_ARTICLE_NUMBER, ADK_PRICE_PRICE_LIST
ADK_DB_TAX_REDUCTION	eTaxReductionPIDInvoice	ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMBER, ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER
ADK_DB_TAX_REDUCTION_ORDER	eTaxReductionPIDOrder	ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMBER, ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER
ADK_DB_TAX_REDUCTION_AVTAL	eTaxReductionPIDAgreement	ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMBER, ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER
ADK_DB_CUSTOMERPAYMENT	eCustomerPaymentInvoice, eCustomerPaymentAdjustment, eCustomerPaymentFee, eCustomerPaymentInvoice-Paydate, eCustomerPaymentNotJournal, eCustomerPaymentAdjustment-Currency, eCustomerPaymentInvoiceRows	ADK_CUSTOMERPAYMENT_INVOICE_NUMBER, ADK_CUSTOMERPAYMENT_ROW_NUMBER

För tabeller som inte räknats upp här fungerar anropet likadant som `AdkFindEx()`.

Nedan följer en tabell med alla sorteringsordningar, till vilka databastabeller de hör och vilket fält som ska anta ett värde vid sökning.

Customer		Urval
eCustomerNr	ADK_CUSTOMER_NUMBER	Alla
eCustomerName	ADK_CUSTOMER_NAME	Alla
eCustomerShortName (1)	ADK_CUSTOMER_SHORT_NAME	Alla
eCustomerPostCode	ADK_CUSTOMER_ZIPCODE	Alla
eCustomerSortOrder (2)	ADK_CUSTOMER_SORT_ORDER	Alla
Offer (1)		
eCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Alla
eCustomerName	ADK_OOI_HEAD_CUSTOMER_NAME	Alla
eOfferNr	ADK_OOI_HEAD_DOCUMENT_NUMBER	Alla
eOfferDate	ADK_OOI_HEAD_DOCUMENT_DATE1	Alla
eOfferNotOrderCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Ej order
eOfferNotPrintedOfferNr	ADK_OOI_HEAD_DOCUMENT_NUMBER	Ej utskrivna
eOfferNotPrintedDate	ADK_OOI_HEAD_DOCUMENT_DATE1	Ej utskrivna
eOfferNotPrintedCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Ej utskrivna
eOfferNotPrintedCustomerName	ADK_OOI_HEAD_CUSTOMER_NAME	Ej utskrivna
Order (2)		
eCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Alla
eCustomerName	ADK_OOI_HEAD_CUSTOMER_NAME	Alla
eOrderNr	ADK_OOI_HEAD_DOCUMENT_NUMBER	Alla
eOrderDate	ADK_OOI_HEAD_DOCUMENT_DATE1	Alla
eOrderDeliveryDate	ADK_OOI_HEAD_DOCUMENT_DATE2	Alla
eOrderNoInvoiceCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Ej Fakturerade
eOrderBackorderOrderNr (1)	ADK_OOI_HEAD_DOCUMENT_NUMBER	Restorder
eOrderNotPrintedOrderNr	ADK_OOI_HEAD_DOCUMENT_NUMBER	Ej utskrivna
eOrderNotPrintedDate	ADK_OOI_HEAD_DOCUMENT_DATE1	Ej utskrivna
eOrderNotPrintedCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Ej utskrivna
eOrderNotPrintedCustomerName	ADK_OOI_HEAD_CUSTOMER_NAME	Ej utskrivna
eOrderNotPrintedDeliveryDate	ADK_OOI_HEAD_DOCUMENT_DATE2	Ej utskrivna
eOrderNoDeliveryNoteOrderNr	ADK_OOI_HEAD_DOCUMENT_NUMBER	Ej följesedel
eOrderNoDeliveryNoteDate	ADK_OOI_HEAD_DOCUMENT_DATE1	Ej följesedel
eOrderNoDeliveryNoteCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Ej följesedel
eOrderNoDeliveryNoteCustomerName	ADK_OOI_HEAD_CUSTOMER_NAME	Ej följesedel
eOrderNoDeliveryNoteDeliveryDate	ADK_OOI_HEAD_DOCUMENT_DATE2	Ej följesedel
eOrderNoInvoice	ADK_OOI_HEAD_DOCUMENT_NUMBER	Ej Fakturerade
eOrderNotDelivered	ADK_OOI_HEAD_DOCUMENT_NUMBER	Ej Levererade
eOrderNotDeliveredCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Ej Levererade
Invoice		
eCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Alla
eCustomerName	ADK_OOI_HEAD_CUSTOMER_NAME	Alla
eInvoiceNr	ADK_OOI_HEAD_DOCUMENT_NUMBER	Alla
eInvoiceDate	ADK_OOI_HEAD_DOCUMENT_DATE1	Alla
eInvoiceInledgerExpireDate	ADK_OOI_HEAD_DOCUMENT_DATE2	I reskontra
eInvoiceInledgerCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	I reskontra
eInvoiceInledgerCustomerName	ADK_OOI_HEAD_CUSTOMER_NAME	I reskontra
eInvoiceInledgerInvoiceNr	ADK_OOI_HEAD_DOCUMENT_NUMBER	I reskontra
eInvoiceNotPayedCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Ej slutbetalda
eInvoiceNotPayedDate	ADK_OOI_HEAD_DOCUMENT_DATE2	Ej slutbetalda
eInvoicePayedCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Slutbetalda
eInvoiceManuellInvoiceNr	ADK_OOI_HEAD_DOCUMENT_NUMBER	Manuella
eInvoiceNotPrintedInvoiceNr	ADK_OOI_HEAD_DOCUMENT_NUMBER	Ej utskrivna
eInvoiceNotPrintedDate	ADK_OOI_HEAD_DOCUMENT_DATE1	Ej utskrivna
eInvoiceNotPrintedCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Ej utskrivna
eInvoiceNotPrintedCustomerName	ADK_OOI_HEAD_CUSTOMER_NAME	Ej utskrivna
eInvoiceNoIntrestInvoiceNr	ADK_OOI_HEAD_DOCUMENT_NUMBER	Ej räntebehandlade
eInvoiceNoIntrestCustomerNr	ADK_OOI_HEAD_CUSTOMER_NUMBER	Ej räntebehandlade
eInvoiceAgreementInvoiceNr (4)	ADK_OOI_HEAD_DOCUMENT_NUMBER	Avtalsfakturer
Supplier Invoice		

eSupplierNr	ADK_SUP_INV_HEAD_SUPPLIER_NUMBER	Alla
eSupplierName	ADK_SUP_INV_HEAD_SUPPLIER_NAME	Alla
eSupplierInvoiceNr	ADK_SUP_INV_HEAD_GIVEN_NUMBER	Alla
eSupplierInvoiceExpireDate	ADK_SUP_INV_HEAD_DUE_DATE	Alla
eSupplierInvoiceNotPaidInvoiceNr	ADK_SUP_INV_HEAD_GIVEN_NUMBER	Ej slutbetalda
eSupplierInvoiceNotPaidSupplierNr	ADK_SUP_INV_HEAD_SUPPLIER_NUMBER	Ej slutbetalda
eSupplierInvoiceNotPaidSupplierName	ADK_SUP_INV_HEAD_SUPPLIER_NAME	Ej slutbetalda
eSupplierInvoiceNotPaidExpireDate	ADK_SUP_INV_HEAD_DUE_DATE	Ej slutbetalda
eSupplierInvoicePaidSupplierNr	ADK_SUP_INV_HEAD_SUPPLIER_NUMBER	Slutbetalda
eSupplierInvoicePaymentAbroadInvoiceNr (3)	ADK_SUP_INV_HEAD_GIVEN_NUMBER	Utländska
eSupplierInvoicePaymentAbroadSupplierNr (3)	ADK_SUP_INV_HEAD_SUPPLIER_NUMBER	Utländska
eSupplierInvoicePaymentAbroadSupplierName (3)	ADK_SUP_INV_HEAD_SUPPLIER_NAME	Utländska
eSupplierInvoicePaymentAbroadExpireDate (3)	ADK_SUP_INV_HEAD_DUE_DATE	Utländska
eSupplierInvoiceDomesticInvoiceNr (3)	ADK_SUP_INV_HEAD_GIVEN_NUMBER	Inhemska
eSupplierInvoiceDomesticSupplierNr (3)	ADK_SUP_INV_HEAD_SUPPLIER_NUMBER	Inhemska
eSupplierInvoiceDomesticSupplierName (3)	ADK_SUP_INV_HEAD_SUPPLIER_NAME	Inhemska
eSupplierInvoiceDomesticExpireDate (3)	ADK_SUP_INV_HEAD_DUE_DATE	Inhemska
eSupplierInvoiceNotJournalInvoiceNr	ADK_SUP_INV_HEAD_GIVEN_NUMBER	Ej journalförda
Article		
eArticleNr	ADK_ARTICLE_NUMBER	Alla
eArticleName	ADK_ARTICLE_NAME	Alla
eArticleShortName (1)	ADK_ARTICLE_SHORT_NAME	Alla
eArticleSortOrder (2)	ADK_ARTICLE_SORT_ORDER	Alla
eArticleGroup	ADK_ARTICLE_GROUP	Alla
eArticleLackOf (2)	ADK_ARTICLE_NUMBER	Brist
eArticlePlaceInStock (2)	ADK_ARTICLE_PLACE_IN_STOCK	Lagervaror
eArticleWebShop	ADK_ARTICLE_WEBSHOP	Webbshopsartikel
eArticleBarCode	ADK_ARTICLE_BAR_CODE	Streckkod
Account		
eAccountNr	ADK_ACCOUNT_NUMBER	Alla
eAccountText	ADK_ACCOUNT_TEXT	Alla
eAccountActiveNr	ADK_ACCOUNT_NUMBER	Aktiva
eAccountActiveText	ADK_ACCOUNT_TEXT	Aktiva
Project (4)		
eProjectCode	ADK_PROJECT_CODE_OF_PROJECT	Alla
eProjectName	ADK_PROJECT_NAME	Alla
eProjectNotFinishedCode	ADK_PROJECT_CODE_OF_PROJECT	Ej avslutade
eProjectNotFinishedName	ADK_PROJECT_NAME	Ej avslutade
eProjectFinishedCode	ADK_PROJECT_CODE_OF_PROJECT	Avslutade
eProjectFinishedName	ADK_PROJECT_NAME	Avslutade
Supplier		
eSupplierNr	ADK_SUPPLIER_NUMBER	Alla
eSupplierName	ADK_SUPPLIER_NAME	Alla
eSupplierShortName (1)	ADK_SUPPLIER_SHORT_NAME	Alla
eSupplierSortOrder (2)	ADK_SUPPLIER_SORT_ORDER	Alla
eSupplierTelephone	ADK_SUPPLIER_TELEPHONE	Alla
Code of Delivery		
eCodeOfTermsOfDelivery	ADK_CODE_OF_TERMS_OF_DELIVERY_CODE	Alla
Way of Delivery		
eCodeOfWayOfDelivery	ADK_CODE_OF_WAY_OF_DELIVERY_CODE	Alla
Code of Payment Terms		

eCodeOfTermsOfPayment	ADK CODE OF TERMS OF PAYMENT CODE	Alla
Code of Language (3)		
eCodeOfLanguage	ADK CODE OF LANGUAGE CODE	Alla
eCodeOfLanguageText	ADK CODE OF LANGUAGE TEXT	Alla
Code of Currency (3)		
eCodeOfCurrency	ADK CODE OF CURRENCY CODE	Alla
Code of Customer Category		
eCodeOfCustomerCategory	ADK CODE OF CUSTOMER CATEGORY CODE	Alla
Code of District (2)		
eCodeOfDistrict	ADK CODE OF DISTRICT CODE	Alla
Code of Seller (2)		
eCodeOfSeller	ADK CODE OF SELLER SIGN	Alla
Discount Agreement		
eDiscountAgreementNr	ADK DISCOUNT AGREEMENT CODE	Alla
eDiscountAgreementText	ADK DISCOUNT AGREEMENT TEXT	Alla
Code of Article Group		
eCodeOfArticleGroup	ADK CODE OF ARTICLE GROUP CODE	Alla
Code of Article Account		
eCodeOfArticleAccount	ADK CODE OF ARTICLE ACCOUNT CODE	Alla
Code of Unit		
eCodeOfUnit	ADK CODE OF UNIT CODE	Alla
eCodeOfUnitLanguage	ADK CODE OF UNIT CODE	Alla
Code of Profit Centre		
eCodeOfProfitCentreUnit	ADK CODE OF PROFIT CENTRE CODE	Alla
Code of Price List		
eCodeOfPriceList	ADK CODE OF PRICE LIST CODE	Alla
Inventory (2)		
eArticleNr	ADK INVENTORY ARTICLE ARTICLE NUMBER	Alla
eArticleName	ADK INVENTORY ARTICLE ARTICLE NAME	Alla
eArticleGroup	ADK INVENTORY ARTICLE ARTICLE GROUP	Alla
eArticlePlaceInStock	ADK INVENTORY ARTICLE ARTICLE PLACE IN STOCK	Lagervaror
eArticleWebShop	ADK ARTICLE WEBSHOP	Webbshopsartikel
eArticleBarCode	ADK ARTICLE BAR CODE	Streckkod
ManualDeliveryIn (2)		
eManualDeliveryIn	ADK MANUAL DELIVERY IN DOCUMENT NUMBER	Alla
ManualDeliveryOut (2)		
eManualDeliveryOut	ADK MANUAL DELIVERY OUT DOCUMENT NUMBER	Alla
Dispatcher (3)		
eDispatcher	ADK DISPATCHER ID	Alla
Booking (1)		
eBookingNr	ADK BOOKING DOCUMENT NUMBER	Alla
eBookingSupplierNumber	ADK BOOKING SUPPLIER NUMBER	Alla
eBookingSupplierName	ADK BOOKING SUPPLIER NAME	Alla
eBookingNotPrintedBookingNr	ADK_BOOKING_DOCUMENT_NUMBER	Ej utskrivna och ej makulerade
eBookingNotPrintedSupplier Number	ADK_BOOKING_SUPPLIER_NUMBER	Ej utskrivna och ej makulerade
eBookingNotPrintedDelivery Date	ADK_BOOKING_DELIVERY_DATE	Ej utskrivna och ej makulerade
eBookingNotDeliveredBookingNr	ADK_BOOKING_DOCUMENT_NUMBER	Ej levererade och utskrivna och ej makulerade
eBookingNotDeliveredSupplier Number	ADK_BOOKING_SUPPLIER_NUMBER	Ej levererade och utskrivna och ej makulerade

eBookingNotDeliveredSupplier Name	ADK_BOOKING_SUPPLIER_NAME	Ej levererade och utskrivna och ej makulerade
eBookingNotDeliveredSupplier DeliveryDate	ADK_BOOKING_DELIVERY_DATE	Ej levererade och utskrivna och ej makulerade
CustomerDiscountRow		
eCustomerDiscountRowArtGrp	ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_CODE	Artikelgruppsrader
eCustomerDiscountRowArtNr	ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_CODE	Artiklarader
ArticleParcel (1)		
eArticleParcelRowNrParcel	ADK_CODE_OF_ARTICLE_PARCEL_NR	Artikel-/textrader
eArticleParcelArticleNrParcel	ADK_CODE_OF_ARTICLE_PARCEL_NR	Artiklarader
eArticleParcelArticleNr	ADK_CODE_OF_ARTICLE_PARCEL_ARTICLE_NR	Artiklarader
ArticleName (3)		
eArticleNameNr	ADK_CODE_OF_ARTICLE_NAME_ARTICLE_NUMBER	Alla
Price		
ePriceArticleNr	ADK_PRICE_ARTICLE_NUMBER	Alla
ePriceBaseArticleNr	ADK_PRICE_ARTICLE_NUMBER	Grundpriser
ePriceNotBaseArticleNr	ADK_PRICE_ARTICLE_NUMBER	Stafflingspriser
ePriceList	ADK_PRICE_PRICE_LIST	Alla
ePriceBaseList	ADK_PRICE_PRICE_LIST	Grundpriser
ePriceNotBaseList	ADK_PRICE_PRICE_LIST	Stafflingspriser
PurchasePrice (2)		
ePurchasePriceArticleNrBase	ADK_ARTICLE_PURCHASE_PRICE_ARTICLE_NUMBER	Grundpriser
ePurchasePriceArticleNrStf	ADK_ARTICLE_PURCHASE_PRICE_ARTICLE_NUMBER	Stafflingspriser
ePurchasePriceArticleNrAll	ADK_ARTICLE_PURCHASE_PRICE_ARTICLE_NUMBER	Alla
ePurchasePriceSupplierNrBase	ADK_ARTICLE_PURCHASE_PRICE_SUPPLIER_NUMBER	Grundpriser
ePurchasePriceSupplierNrStf	ADK_ARTICLE_PURCHASE_PRICE_SUPPLIER_NUMBER	Stafflingspriser
ePurchasePriceSupplierArticleNr	ADK_ARTICLE_PURCHASE_PRICE_SUPPLIER_NUMBER	Alla
WayOfPayment		
eWayOfPayment		Alla
Member (5)		
eMemberNr	ADK_MEMBER_NUMBER	Alla
eMemberFirstName	ADK_MEMBER_FIRST_NAME	Alla
eMemberLastName	ADK_MEMBER_LAST_NAME	Alla
eMemberPostCode	ADK_MEMBER_ZIPCODE	Alla
eMemberPersonalNumber	ADK_MEMBER_PERSONAL_NUMBER	Alla
eMemberCategory	ADK_MEMBER_CATEGORY	Alla
eMemberFreeCategory1	ADK_MEMBER_FREE_CATEGORY1	Alla
eMemberFreeCategory2	ADK_MEMBER_FREE_CATEGORY2	Alla
eMemberFreeCategory3	ADK_MEMBER_FREE_CATEGORY3	Alla
eMemberFreeCategory4	ADK_MEMBER_FREE_CATEGORY4	Alla
eMemberFreeCategory5	ADK_MEMBER_FREE_CATEGORY5	Alla
eMemberFreeCategory6	ADK_MEMBER_FREE_CATEGORY6	Alla
eMemberFreeCategory7	ADK_MEMBER_FREE_CATEGORY7	Alla
eMemberFreeCategory8	ADK_MEMBER_FREE_CATEGORY8	Alla
eMemberFreeCategory9	ADK_MEMBER_FREE_CATEGORY9	Alla
eMemberFreeCategory10	ADK_MEMBER_FREE_CATEGORY10	Alla
DeliveryNote (1)		
eDeliveryNoteNr	ADK_DELIVERY_NOTE_DOCUMENT_NUMBER	Alla
eDeliveryNoteSupplierNumber	ADK_DELIVERY_NOTE_SUPPLIER_NUMBER	Alla
eDeliveryNoteSupplierName	ADK_DELIVERY_NOTE_SUPPLIER_NAME	Alla
eDeliveryNoteInvoiceNotSentDeliveryNoteNr	ADK_DELIVERY_NOTE_DOCUMENT_NUMBER	Ej fakturerade
eDeliveryNoteInvoiceNotSentSupplierNumber	ADK_DELIVERY_NOTE_SUPPLIER_NUMBER	Ej fakturerade
eDeliveryNoteInvoiceNotSentSupplierName	ADK_DELIVERY_NOTE_SUPPLIER_NAME	Ej fakturerade
Package (2)		
ePackageOrderNr	ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT	Alla
PackagelImport (1)		
eImpPackageDelNoteNr	ADK_IMP_PACKAGE_HEAD_CONNECTION_DOCUMENT	Alla

DeliveryAddress		
eDeliveryAddressCustomerNr	ADK DELIVERY CUSTOMER NUMBER	Alla
YourRefCustomer		
eCodeOfYourRefCustomer	ADK CODE OF YOUR REF CUSTOMER NUMBER	Alla
YourRefSupplier		
eCodeOfYourRefSupplier	ADK CODE OF YOUR REF SUPPLIER NUMBER	Alla
CodeOfCountry		
eCodeOfCountryCode	ADK CODE OF COUNTRY CODE CODE	Alla
CustomerPayment		
eCustomerPaymentInvoice	ADK CUSTOMERPAYMENT INVOICE NUMBER	Alla
eCustomerPaymentAdjustment	ADK CUSTOMERPAYMENT INVOICE NUMBER	Bortskrivningar
eCustomerPaymentFee	ADK CUSTOMERPAYMENT INVOICE NUMBER	Avgifter
eCustomerPaymentInvoicePaydate	ADK CUSTOMERPAYMENT INVOICE NUMBER	Alla
eCustomerPaymentNotJournal	ADK CUSTOMERPAYMENT INVOICE NUMBER	Ej journalförda
eCustomerPaymentAdjustmentCurrency	ADK_CUSTOMERPAYMENT_INVOICE_NUMBER	Kursdiffar
eCustomerPaymentInvoiceRows	ADK CUSTOMERPAYMENT INVOICE NUMBER	Alla
eCustomerPaymentDate	ADK CUSTOMERPAYMENT DATE	Alla
CodeOfAdjustment		
eCodeOfAdjustmenCode	ADK CODE OF ADJUSTMENT CODE CODE	Alla
SupplierPayment		
eSupplierPaymentAll	ADK SUPPLIERPAYMENT PAYDATE	Alla
eSupplierPaymentInvAll	ADK SUPPLIERPAYMENT INVOICE NR	Alla
eSupplierPaymentInvNotPayed	ADK SUPPLIERPAYMENT INVOICE NR	Ej betalda
eSupplierPaymentInvOnWay	ADK SUPPLIERPAYMENT INVOICE NR	På väg
eSupplierPaymentInvPayed	ADK SUPPLIERPAYMENT INVOICE NR	Betalda
CashReceipt		
eCashReceiptNr	ADK OOI HEAD DOCUMENT NUMBER	Alla
eCashReceiptDate	ADK OOI HEAD DOCUMENT DATE1	Alla
eCashReceiptCustomerNr	ADK OOI HEAD CUSTOMER NUMBER	Alla
eCashReceiptCustomerName	ADK OOI HEAD CUSTOMER NAME	Alla
Verification		
eVerificationNr	ADK VERIFICATION HEAD VER NR	Alla
eVerificationTDate	ADK VERIFICATION HEAD DATE1	Alla
Bookingyear		
eCodeOfBookingyear	ADK CODE OF BOOKINGYEAR ID	Alla
eCodeOfBookingyearStart	ADK CODE OF BOOKINGYEAR START	Alla
DiscountCode		
eCodeOfDiscount	ADK_CODE_OF_DISCOUNT_FILTER, ADK_CODE_OF_DISCOUNT_CODE	Alla
Contact (3)		
eContactID	ADK CONTACTS ID	Alla
eContact	ADK CONTACTS NAME	Alla
eContactCompany	ADK CONTACTS COMPANY NAME	Alla
eContactTitle	ADK CONTACTS TITLE	Alla
eContactCustomer	ADK CONTACTS NAME	Kopplade kunder
eContactCustomerCompany	ADK CONTACTS COMPANY NAME	Kopplade kunder
eContactCustomerTitle	ADK CONTACTS TITLE	Kopplade kunder
eContactCustomerNr	ADK_CONTACTS_CUSTOMER_NUMBER, ADK CONTACTS NAME	Kopplade kunder
eContactSupplier	ADK_CONTACTS_NAME	Kopplade leverantör
eContactSupplierCompany	ADK_CONTACTS_COMPANY_NAME	Kopplade leverantör
eContactSupplierTitle	ADK_CONTACTS_TITLE	Kopplade leverantör
eContactSupplierNr	ADK CONTACTS SUPPLIER_NUMBER, ADK CONTACTS NAME	Kopplade leverantör
eContactOther	ADK CONTACTS NAME	Ej kopplade
eContactOtherCompany	ADK CONTACTS COMPANY NAME	Ej kopplade

eContactOtherTitle	ADK CONTACTS TITLE	Ej kopplade
eContactCommercialSMS	ADK CONTACTS NAME	Godkänd SMS
eContactCommercialEmail	ADK CONTACTS NAME	Godkänd Epost
eContactCustomerSign	ADK CONTACTS CUSTOMER_NUMBER, ADK CONTACTS SIGN	Kopplade kund
eContactSupplierSign	ADK CONTACTS SUPPLIER_NUMBER, ADK CONTACTS SIGN	Kopplade leverantörer
eContactOtherSign	ADK CONTACTS SIGN	Ej kopplade
ContactTitle (3)		
eCodeOfContactTitle	ADK CODE OF CONTACT TITLE ID	Alla
eCodeOfContactTitleName	ADK CODE OF CONTACT TITLE NAME	Alla
ContactGroup (3)		
eCodeOfContactGroup	ADK CODE OF CONTACT GROUPS ID	Alla
eCodeOfContactGroupName	ADK CODE OF CONTACT GROUPS NAME	Alla
ContactConnection (3)		
eCodeOfContactsAndContactsGroup	ADK_CODE_OF_CONTACT_GROUP_CONTACTS_CONTACT ID	Alla
eCodeOfContactsAndContactsGroupGroupID	ADK_CODE_OF_CONTACT_GROUP_CONTACTS_GROUP_I D	Alla
TaxReduction (3)		
eTaxReductionInvoiceNumber	ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMB ER	Alla
eTaxReductionInvoiceNumberApplied	ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMB ER	Ansökta
eTaxReductionInvoiceNumberGranted	ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMB ER	Beviljade
eTaxReductionPersonalIdentityNumber	ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER	Alla
eTaxReductionPersonalIdentityNumberApplied	ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER	Ansökta
eTaxReductionPersonalIdentityNumberGranted	ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER	Beviljade
eTaxReductionPIDInvoice	ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMB ER, ADK TAX REDUCTION PERSONAL IDENTITY NUMBER	Alla
Agreement (4)		
eAgreementNr	ADK AGREEMENT HEAD DOCUMENT NUMBER	Alla
eCustomerNr	ADK AGREEMENT HEAD CUSTOMER NUMBER	Alla
eCustomerName	ADK AGREEMENT HEAD CUSTOMER NAME	Alla
TaxReductionOrder (2)		
eTaxReductionOrderNumber	ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMB ER	Alla
eTaxReductionPersonalIdentityNumberOrder	ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER	Alla
eTaxReductionPIDOrder	ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMB ER	Alla
TaxReduktionAvtal (2)		
eTaxReductionAgreementNumber	ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMB ER	Alla
eTaxReductionPersonalIdentityNumberAgreement	ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER	Alla
eTaxReductionPIDAgreement	ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMB ER	Alla
VerificationSeries		
eVerSerie	ADK VERIFICATION SERIES VER SERIE	Alla
BookKeepingHistory		
eBookkeepingHistNr	ADK BOOKKEEPINGHIST NR	Alla
eBookkeepingHistDate	ADK BOOKKEEPINGHIST DATE	Alla
eBookkeepingHistText	ADK BOOKKEEPINGHIST TEXT	Alla
Free Category (5)		
eFreeCategory	ADK FREE CATEGORY CATEGORY	Alla
Periodic adjustment		

ePeriodicAdjustmentAccount	ADK_PERIODIC_ADJUSTMENT_ACCOUNT	Alla
CustomerArticle		
eCustomerArticleeArtnrCustnr	ADK_CUSTOMER_ARTICLE_ARTNR	Alla
eCustomerArticleCustnrArtnr	ADK_CUSTOMER_ARTICLE_CUSTNR	Alla
eCustomerArticleCustnrCustartnr	ADK_CUSTOMER_ARTICLE_CUSTNR	Alla
AttachmentInfo		
eAttachmentInfoSupplierInvoice	ADK_ATTACHMENT_INFO_DOKNR	Kopplade till levfkt
eAttachmentInfoVerification	ADK_ATTACHMENT_INFO_VERNR	Kopplade till ver
TaxReductionTypes (3)		
eTaxReductionTypesId	ADK_TAX_REDUCTION_TYPES_ID	Alla
Customer		
eCustomerEmail	ADK_CUSTOMER_EMAIL	Alla

- (1) Registret/sökordningen används endast i Visma Administration 2000.
(2) Registret/sökordningen används endast i Visma Administration 1000/2000.
(3) Registret/sökordningen används endast i Visma Administration 500/1000/2000.
(4) Registret/sökordningen används endast i Visma Administration 1000/2000 och Visma Förening.
(5) Registret/sökordningen används endast i Visma Förening

Sker en sökning med kundnamn som vald sorteringsordning kommer det värde som finns i fältet för kundnamn att fungera som sökvärde. Saknar detta fält värde, returneras ett felmeddelande. Den post som returneras vid träff är den första för aktuellt sökvärde. Skulle det finnas fler poster som motsvarar sökvärdet kan funktionen `AdkNext()` användas för att gå framåt till nästa post.

Sökningen fungerar olika för tabeller med numeriskt värde i nyckelfältet och tabeller med text i nyckelfältet. För numeriska värden krävs exakt rätt värde för träff, men för text räcker det med att inledningen av nyckelvärdet motsvarar hela det sökvärde som angetts.

Exempel:

Kundregistret innehåller 2 poster med kundnr "101" resp "110". En sökning på "11" ger träff på kund "110" och ADKE_OK returneras. Om exakt värde söks, kan det vara lämpligt att göra en jämförelse mellan träffat värde och det som söktes, för att säkerställa att exakt rätt post hittats.

Om sökning sker på order, faktura, offert eller leverantörsfaktura när dessa innehåller kopplingar mot dokumentrader, kommer även en datastruktur för de tillhörande dokumentraderna att skapas. Såväl dokumenthuvudet som dokumentraderna fylls i med data. Detta innebär att det är möjligt att arbeta med alla de rader som tillhör en post. `AdkFind()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av anropet. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkFirst

C++

```
ADK_ERROR AdkFirst(ADK_DATA* pData)
```

C#

```
ADKERROR AdkFirst(int pDb)
```

Information

Funktionen `AdkFirst()` hämtar den första posten i en databastabell. Funktionen tar en inparameter, en pekare till en datastruktur. Denna datastruktur kommer att fyllas med data från den första posten.

Denna funktion kan användas på alla de programdelar som anges i tabell 1. Funktionen `AdkFirst()` utgår från aktuell sorteringsordning för databastabellen. Om databastabellen för kunder för närvarande

har sorteringsordningen "kundnamn" kommer den första posten utifrån denna sorteringsordning att hämtas. Resultatet av funktionen `AdkFirst()` varierar beroende på vilken sorteringsordning som är aktiv för en databastabell.

Om sökning sker på order, faktura, offert eller leverantörsfaktura när dessa innehåller kopplingar mot andra dokumentrader, kommer även en datastruktur för de tillhörande dokumentraderna att skapas. Såväl dokumenthuvudet som dokumentraderna fylls i med data. Detta innebär att det är möjligt att arbeta med alla de rader som tillhör en post. `AdkFirst()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av anropet. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkNext

C++

```
ADK_ERROR AdkNext(ADK_DATA* pData)
```

C#

```
ADKERROR AdkNext(int pDb)
```

Information

Funktionen `AdkNext()` hämtar nästa post i en databastabell. Funktionen tar en inparameter, en pekare till en datastruktur. Denna datastruktur kommer att fyllas med data från nästa post räknat från den aktuella postens innehåll. En förutsättning för att använda denna funktion är att den datastruktur som skickas in till funktionen innehåller en post. Det är utifrån denna post som nästa post kommer att hämtas. För att få en post till datastrukturen kan t ex någon av funktionerna `AdkFind()` eller `AdkFirst()` först anropas. Därefter kan `AdkNext()` anropas.

Denna funktion kan användas på alla de programdelar som anges i tabell 1. Funktionen `AdkNext()` utgår från aktuell sorteringsordning för databastabellen. Om databastabellen för kunder för närvarande har sorteringsordningen "Kundnamn" kommer nästa post utifrån denna sorteringsordning att hämtas in. Resultatet av funktionen `AdkNext()` varierar beroende på vilken sorteringsordning som är aktiv för en databastabell.

Om sökning sker på order, faktura, offert eller leverantörsfaktura när dessa innehåller kopplingar mot andra dokumentrader, kommer även en datastruktur för de tillhörande dokumentraderna att skapas. Såväl dokumenthuvudet som dokumentraderna fylls i med data. Detta innebär att det är möjligt att arbeta med alla de rader som tillhör en post. `AdkNext()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkPrevious

C++

```
ADK_ERROR AdkPrevious(ADK_DATA* pData)
```

C#

```
ADKERROR AdkPrevious(int pDb)
```

Information

Funktionen `AdkPrevious()` hämtar föregående post i en databastabell. Funktionen tar en inparameter, en pekare till en datastruktur. Denna datastruktur kommer att fyllas med data från föregående post räknat från den aktuella postens innehåll. En förutsättning för att använda denna funktion är att den datastruktur som skickas in till funktionen innehåller en post. Det är utifrån denna post som föregående post kommer att hämtas. För att få en post till datastrukturen kan t ex någon av funktionerna `AdkFind()` eller `AdkLast()` först anropas. Därefter kan `AdkPrevious()` anropas.

Denna funktion kan användas på alla de programdelar som anges i tabell 1. Funktionen `AdkPrevious()` utgår från aktuell sorteringsordning för databastabellen. Om databastabellen för kunder för närvarande har sorteringsordningen "Kundnamn" kommer föregående post utifrån denna sorteringsordning att hämtas in. Resultatet av funktionen `AdkPrevious()` varierar beroende på vilken sorteringsordning som är aktiv för en databastabell.

Om sökning sker på order, faktura, offert eller leverantörsfaktura när dessa innehåller kopplingar mot andra dokumentrader, kommer även en datastruktur för de tillhörande dokumentraderna att skapas. Såväl dokumenthuvudet som dokumentraderna fylls i med data. Detta innebär att det är möjligt att arbeta med alla de rader som tillhör en post. `AdkPrevious()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkLast

C++

```
ADK_ERROR AdkLast(ADK_DATA* pData)
```

C#

```
ADKERROR AdkLast(int pDb)
```

Information

Funktionen `AdkLast()` hämtar den sista posten i en databastabell. Funktionen tar en inparameter, en pekare till en datastruktur. Denna datastruktur kommer att fyllas med data från den sista posten.

Denna funktion kan användas på alla de programdelar som anges i tabell 1. Funktionen `AdkLast()` utgår från aktuell sorteringsordning för databastabellen. Om databastabellen för kunder för närvarande har sorteringsordningen "kundnamn" kommer den sista posten utifrån denna sorteringsordning att hämtas. Resultatet av funktionen `AdkLast()` varierar beroende på vilken sorteringsordning som är aktiv för en databastabell.

Om sökning sker på order, faktura, offert eller leverantörsfaktura när dessa innehåller kopplingar mot andra dokumentrader, kommer även en datastruktur för de tillhörande dokumentraderna att skapas. Såväl dokumenthuvudet som dokumentraderna fylls i med data. Detta innebär att det är möjligt att arbeta med alla de rader som tillhör en post. `AdkLast()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av anropet. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkNextEx, AdkPreviousEx, AdkFirstEx, AdkLastEx

C++

```
ADK_ERROR AdkNextEx(ADK_DATA* pData, BOOL bRow)
ADK_ERROR AdkPreviousEx(ADK_DATA* pData, BOOL bRow)
ADK_ERROR AdkFirstEx(ADK_DATA* pData, BOOL bRow)
ADK_ERROR AdkLastEx(ADK_DATA* pData, BOOL bRow)
```

C#

```
ADKERROR AdkNextEx(int pDb, bool bRow)
ADKERROR AdkPreviousEx(int pDb, bool bRow)
ADKERROR AdkFirstEx(int pDb, bool bRow)
ADKERROR AdkLastEx(int pDb, bool bRow)
```

Information

Fungerar exakt som sina motsvarigheter med tillägget att det går att välja om rader ska läsas in eller inte om sådana finns (ex på fakturor). Detta är gjort för att det ska gå snabbare att söka igenom tabeller med kopplade rader. Effekten blir tydligare när databasanrop går lite långsammare, tex över nätverk. Det är den andra parametern i anropet som bestämmer huruvida rader ska tas med eller inte. Värdet TRUE anger att raderna ska tas med i svaret. Värdet FALSE anger att raderna inte ska tas med i svaret. (TRUE = 1, FALSE = 0 enligt definition i windows headerfiler). Om rader inte tas med hämtas således endast huvudet.

AdkNextEx(), AdkPreviousEx(), AdkFirstEx(), AdkLastEx() finns i AdkNet4Wrapper i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkUpdate

C++

```
ADK_ERROR AdkUpdate(ADK_DATA* pData)
```

C#

```
ADKERROR AdkUpdate(int pDb)
```

Information

Funktionen AdkUpdate() uppdaterar en befintlig post i databasen. För att peka ut vilken post i en databastabell som ska uppdateras används den aktuella databastabellens nyckel. Funktionen kan ej användas på alla deklarerade typer av datastrukturer. Nedan följer en lista på de typer som AdkUpdate() kan utföras på. I listan framgår även vilken nyckel som respektive register har:

- Offert – ADK_OOI_HEAD_DOCUMENT_NUMBER
- Order – ADK_OOI_HEAD_DOCUMENT_NUMBER
- Kundfaktura (normalfaktura, kontantnota och kreditfaktura) – ADK_OOI_HEAD_DOCUMENT_NUMBER.
- Leverantörsfaktura (normalfaktura och kreditfaktura) – ADK_SUP_INV_HEAD_GIVEN_NUMBER

- Inkommande följesedel – ADK_DELIVERY_NOTE_DOCUMENT_NUMBER
- Beställning – ADK_BOOKING_DOCUMENT_NUMBER
- Kund – ADK_CUSTOMER_NUMBER.
- Medlem – ADK_MEMBER_NUMBER
- Artikel – ADK_ARTICLE_NUMBER
- Leverantör – ADK_SUPPLIER_NUMBER
- Projekt – ADK_PROJECT_CODE_OF_PROJECT
- Inventering – ADK_INVENTORY_ARTICLE_ARTICLE_NUMBER
- Manuell inleverans – ADK_MANUAL_DELIVERY_IN_DOCUMENT_NUMBER
- Manuell utleverans – ADK_MANUAL_DELIVERY_OUT_DOCUMENT_NUMBER
- Säljare – ADK_CODE_OF_SELLER_SIGN
- Distrikt – ADK_CODE_OF_DISTRICT_CODE
- Kundkategori – ADK_CODE_OF_CUSTOMER_CATEGORY_CODE
- Betalningsvillkor – ADK_CODE_OF_TERMS_OF_PAYMENT_CODE
- Leveransvillkor – ADK_CODE_OF_TERMS_OF_DELIVERY_CODE
- Leveranssätt – ADK_CODE_OF_WAY_OF_DELIVERY_CODE
- Fri kategori 1 .. Fri kategori 10 – ADK_FREE_CATEGORY_CATEGORY
- Kolli – ADK_PACKAGE_HEAD_PACKAGE_NUMBER,
ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT
- Avvikande leveransadress – ADK_DELIVERY_CUSTOMER_NUMBER,
ADK_DELIVERY_NAME

- Er referens, kunder – ADK_CODE_OF_YOUR_REF_CUSTOMER_NUMBER,
ADK_CODE_OF_YOUR_REF_SIGN
- Er referens, leverantörer – ADK_CODE_OF_YOUR_REF_SUPPLIER_NUMBER,
ADK_CODE_OF_YOUR_REF_SUPPLIER_SIGN
- Landskoder – ADK_CODE_OF_COUNTRY_CODE_CODE
- Bortskrivningskoder – ADK_CODE_OF_ADJUSTMENT_CODE_CODE
- Valuta – ADK_CODE_OF_CURRENCY_CODE
- Artikelgrupper – ADK_CODE_OF_ARTICLE_GROUP_CODE
- Enhetskoder – ADK_CODE_OF_UNIT_CODE
- Resultatenheter – ADK_CODE_OF_PROFIT_CENTRE_CODE
- Inköspriser – ADK_ARTICLE_PURCHASE_PRICE_ARTICLE_NUMBER
- Betalsätt – ADK_CODE_OF_WAY_OF_PAYMENT_CODE
- Kontakt – ADK_CONTACTS_ID
- Kontaktgrupper – ADK_CODE_OF_CONTACT_GROUPS_ID
- Kontakttitlar – ADK_CODE_OF_CONTACT_TITLE_ID
- Skattereduktion (husarbete) – ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMBER,
ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER
- Artikelbenämning – ADK_CODE_OF_ARTICLE_NAME_ARTICLE_NUMBER,
ADK_CODE_OF_ARTICLE_NAME_LANGUAGE_CODE
- Rabattavtal – ADK_DISCOUNT_AGREEMENT_CODE
- Rabattavtalsrad – ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_CODE,

ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_ARTICLE_GROUP eller
ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_ARTICLE_NUMBER

- **Språk** – ADK_CODE_OF_LANGUAGE_CODE
- **Pris** – ADK_PRICE_ARTICLE_NUMBER,
ADK_PRICE_PRICE_LIST
- **Prislista** – ADK_CODE_OF_PRICE_LIST_CODE
- **Speditör** – ADK_DISPATCHER_ID
- **Avtal** – ADK_AGREEMENT_HEAD_DOCUMENT_NUMBER
- **Konto** – ADK_ACCOUNT_NUMBER
- **Kundartiklar** – ADK_CUSTOMER_ARTICLE_CUSTNR, ADK_CUSTOMER_ARTICLE_ARTNR

De nyckelfält som anges ovan är de som ska användas för att peka ut vilken post som ska uppdateras. Detta sker genom att aktuellt fält tilldelas ett värde som pekar ut önskad post. När det gäller order, offert, faktura och leverantörsfaktura är det även möjligt att uppdatera tillhörande rader samt att skapa nya rader till ett befintligt dokumenthuvud. Hur rader uppdateras och skapas beskrivs mer utförligt i bilagan **Kodbeskrivning**. Det är ej möjligt att uppdatera de nyckelfält som beskrivits ovan.

För att utföra en uppdatering med `AdkUpdate()` anges först ett nyckelvärde för att peka ut vilken post som ska uppdateras. När nyckelvärde är satt anges de värden som ska uppdateras och sedan anropas funktionen `AdkUpdate()` för att utföra själva uppdateringen. Det är även möjligt att uppdatera en post utan att ange ett nyckelvärde. Har en post hämtats med `AdkFind()`, `AdkFirst()`, `AdkLast()`, `AdkNext()` eller `AdkPrevious()` så uppdateras denna post när `AdkUpdate()` anropas. `AdkUpdate()` kontrollerar de värden som angetts. Om alla värden är giltiga kommer den utpekade posten i databasen att uppdateras. Endast fält med där flaggan `bChanged` satt till sant, d v s de fält som tilldelats värden med någon av `AdkSet`-funktionerna, kommer att uppdateras. När uppdateringen har skett återställs flaggan `bChanged`.

När en rad på en leverantörsfaktura ska uppdateras är det inte säkert att raden har samma radnummer som när den skapades. Vid skapandet av en leverantörsfaktura kan det tillkomma upp till tre rader beroende på omfälten för totalt, frakt och moms har värden skilda från noll. När dessa fält har antagit värden, kommer en rad att skapas för vart och ett av dessa fälten. Dessa rader får då nummer ett till tre. Har värden satts för tex totalt och moms, men inte för frakt, kommer dessa två rader automatiskt att ligga först. Som rad 3 kommer då den första raden du själv lagt till i leverantörsfakturan.

`AdkUpdate()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkUpdateEx

C++

```
ADK_ERROR AdkUpdateEx(ADK_DATA* pData, BOOL bRow)
```

C#

```
ADKERROR AdkUpdateEx(int pDb, bool bRow)
```

Information

Fungerar exakt som sin motsvarighet med tillägget att det går att välja om rader ska läsas in eller inte om sådana finns (ex på fakturor) efter att operationen ifråga utförts. Detta är gjort för att det ska gå snabbare att uppdatera poster i tabeller med kopplade rader. Effekten blir tydligare när databasanrop går lite långsammare, tex över nätverk. Det är den andra parametern i anropet som bestämmer huruvida rader ska tas med eller inte. Värdet `TRUE` anger att raderna ska tas med i svaret. Värdet `FALSE` anger att raderna inte ska tas med i svaret. (`TRUE` = 1, `FALSE` = 0 enligt definition i windows headerfiler). Om rader inte tas med hämtas således endast huvudet.

AdkUpdateEx() finns i *AdkNet4Wrapper* i två varianter. Den äldre har deklarerats som *deprecated* i *.NET*. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i *.NET*. Dock är *AdkNet4Wrapper.dll* bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkDelete/AdkDeleteRecord

C++

```
ADK_ERROR AdkDelete(ADK_DATA* pData)  
ADK_ERROR AdkDeleteRecord(ADK_DATA* pData)
```

C#

```
ADKERROR AdkDeleteRecord(int pDb)
```

Information

Funktionen `AdkDelete()` raderar en post i databasen. För att peka ut vilken post som ska raderas används det aktuella registrets nyckel, t ex artikelnummer om det är en artikelpost som ska raderas. Funktionen tar en inparameter, pekaren till den datastruktur där nyckeln för den post som ska raderas är angiven. Nedan följer en lista över de programdelar där det är möjligt att utföra `AdkDelete()` samt den nyckel som ska användas för att peka ut vilken post som ska raderas:

- Kund – `ADK_CUSTOMER_NUMBER`
- Medlem – `ADK_MEMBER_NUMBER`
- Artikel – `ADK_ARTICLE_NUMBER`
- Leverantör – `ADK_SUPPLIER_NUMBER`
- Projekt – `ADK_PROJECT_CODE_OF_PROJECT`
- Manuell inleverans – `ADK_MANUAL_DELIVERY_IN_DOCUMENT_NUMBER`
- Manuell utleverans – `ADK_MANUAL_DELIVERY_OUT_DOCUMENT_NUMBER`

- **Säljare** – ADK_CODE_OF_SELLER_SIGN
- **Distrikt** – ADK_CODE_OF_DISTRICT_CODE
- **Kundkategori** – ADK_CODE_OF_CUSTOMER_CATEGORY_CODE
- **Betalningsvillkor** – ADK_CODE_OF_TERMS_OF_PAYMENT_CODE
- **Leveransvillkor** – ADK_CODE_OF_TERMS_OF_DELIVERY_CODE
- **Leveranssätt** – ADK_CODE_OF_WAY_OF_DELIVERY_CODE
- **Fri kategori 1 .. Fri kategori 10** – ADK_FREE_CATEGORY_CATEGORY
- **Kolli** – ADK_PACKAGE_HEAD_PACKAGE_NUMBER
ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT
- **Avvikande leveransadress** – ADK_DELIVERY_CUSTOMER_NUMBER,
ADK_DELIVERY_NAME
- **Er referens, kunder** – ADK_CODE_OF_YOUR_REF_CUSTOMER_NUMBER,
ADK_CODE_OF_YOUR_REF_SIGN
- **Er referens, leverantörer** – ADK_CODE_OF_YOUR_REF_SUPPLIER_NUMBER,
ADK_CODE_OF_YOUR_REF_SUPPLIER_SIGN
- **Landskoder** – ADK_CODE_OF_COUNTRY_CODE_CODE
- **Bortskrivningskoder** – ADK_CODE_OF_ADJUSTMENT_CODE_CODE
- **Valuta** – ADK_CODE_OF_CURRENCY_CODE
- **Artikelgrupper** – ADK_CODE_OF_ARTICLE_GROUP_CODE
- **Enhetskoder** – ADK_CODE_OF_UNIT_CODE
- **Resultatenheter** – ADK_CODE_OF_PROFIT_CENTRE_CODE
- **Inköpspriser** – ADK_ARTICLE_PURCHASE_PRICE_ARTICLE_NUMBER

- **Betalsätt** – ADK_CODE_OF_WAY_OF_PAYMENT_CODE
- **Kundinbetalningar** – ADK_CUSTOMERPAYMENT_INVOICE_NUMBER,
ADK_CUSTOMERPAYMENT_ROW_NUMBER
- **Kontakt** – ADK_CONTACTS_ID
- **Kontaktgrupper** – ADK_CODE_OF_CONTACT_GROUPS_ID
- **Kontakttitlar** – ADK_CODE_OF_CONTACT_TITLE_ID
- **Kontaktkopplingar** – ADK_CODE_OF_CONTACT_GROUP_CONTACTS_CONTACT_ID,
ADK_CODE_OF_CONTACT_GROUP_CONTACTS_GROUP_ID
- **Skattereduktion (husarbete)** – ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMBER,
ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER
- **Artikelbenämning** – ADK_CODE_OF_ARTICLE_NAME_ARTICLE_NUMBER,
ADK_CODE_OF_ARTICLE_NAME_LANGUAGE_CODE
- **Rabattavtal** – ADK_DISCOUNT_AGREEMENT_CODE
- **Rabattavtalsrad** – ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_CODE,
ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_ARTICLE_GROUP **eller**
ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_ARTICLE_NUMBER
- **Språk** – ADK_CODE_OF_LANGUAGE_CODE
- **Pris** – ADK_PRICE_ARTICLE_NUMBER,
ADK_PRICE_PRICE_LIST
- **Prislista** – ADK_CODE_OF_PRICE_LIST_CODE
- **Speditör** – ADK_DISPATCHER_ID
- **Konto** – ADK_ACCOUNT_NUMBER
- **Kundartiklar** – ADK_CUSTOMER_ARTICLE_CUSTNR, ADK_CUSTOMER_ARTICLE_ARTNR
eller ADK_CUSTOMER_ARTICLE_CUSTNR, ADK_CUSTOMER_ARTICLE_CUSTARTNR

Observera att det inte endast är de värden som finns i den datastruktur som skickas in till `AdkDelete()` som kommer att raderas utan hela posten i databasen tas bort.

När raderingen är utförd avallokerar `AdkDelete()` det minne som tidigare allokerats till datastrukturen. `AdkDelete()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

OBS! `pData` sätts inte till `NULL` av `AdkDelete()`. Minnesfel kan uppstå ifall innehållet i `pData` försöker användas när `AdkDelete()` returnerat `ADKE_OK`. Returneras något annat ska `AdkDeleteStruct()` anropas för att avallokera innehållet i `pData`.

`AdkDeleteRecord()` har tillkomit för att beteendet ska vara så likt som möjligt jämfört med andra funktioner. Funktionen `AdkDeleteRecord()` avallokerar inte minnet som används och ska således, oavsett returvärdet, `AdkDeleteStruct()` anropas för att avallokera innehållet i `pData`.

`AdkDelete()` har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och funktionen kommer fortsätta att fungera som innan.

AdkDeleteRow

C++

```
ADK_ERROR AdkDeleteRow(ADK_DATA* pData, INT iRowNr)
```

C#

```
ADKERROR AdkDeleteRow(int pDb, int iIx)
```

Information

`AdkDeleteRow()` raderar en rad i en datastruktur. För att peka ut i vilken post en rad ska raderas används det aktuella registrets nyckel. Nyckelfältet i datastrukturen tilldelas ett värde som sedan kommer att fungera som sökvärde för att hitta den post där en tillhörande rad ska tas bort. Radering av rader är endast aktuellt för de datastrukturer som kan ha tillhörande rader. Nedan följer en lista över de programdelar där det är möjligt att utföra `AdkDeleteRow()` samt den nyckel som ska användas för att peka ut den post i vilken en tillhörande rad ska raderas:

- Order – `ADK_OOI_HEAD_DOCUMENT_NUMBER`.
- Offert – `ADK_OOI_HEAD_DOCUMENT_NUMBER`.
- Faktura – `ADK_OOI_HEAD_DOCUMENT_NUMBER`.
- Leverantörsfaktura – `ADK_SUP_INV_HEAD_GIVVEN_NUMBER`.
- Inkommande följesedel – `ADK_DELIVERY_NOTE_DOCUMENT_NUMBER`.
- Beställning – `ADK_BOOKING_DOCUMENT_NUMBER`.
- Kolli – `ADK_PACKAGE_HEAD_PACKAGE_NUMBER`
`ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT`

- Avtal – ADK_AGREEMENT_HEAD_DOCUMENT_NUMBER

Det är endast möjligt att radera en rad i taget med hjälp av funktionen `AdkDeleteRow()`.

`AdkDeleteRow()` tar två inparametrar. Den första inparametern är en pekare till det dokumenthuvud som innehåller den rad som ska tas bort. Den andra inparametern anger numret på den rad som ska tas bort. Första raden har nummer 1, andra har nummer 2 osv. Om t ex rad nummer sju ska tas bort ska alltså 7 anges. Om ytterligare rader ska raderas måste funktionen anropas igen med en ny inparameter. Observera att numret måste justeras i förhållande till den rad som togs bort. Om t ex rad nummer 7 togs bort första gången har numreringen ändrats och rad nummer 10 har nu fått nummer 9 osv.

Om raderingen av raden går bra kommer all data som tillhör aktuell post att läsas upp i dokumenthuvudet, det kommer att skapas rader efter det nya antalet och dessa kommer att fyllas med data.

`AdkDeleteRow()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

Hjälpfunktioner som ej utför databasoperationer

Här beskrivs de funktioner som inte arbetar direkt mot databasen men ändå behövs för att API:et ska fungera korrekt.

AdkOpen

C++

```
ADK_ERROR AdkOpen (CHAR* pszSystemPath, CHAR* pszFtgPath)
```

Information

Funktionen `AdkOpen()` används för att öppna en databas till ett befintligt företag eller förening. Funktionen tar två inparametrar, den första är en sträng som pekar ut var **Visma Administration 500/1000/2000** respektive **Visma Förening** är installerade och den andra är en sträng som pekar ut var önskat företag eller förening finns. När funktionen `AdkOpen()` anropas kontrolleras de licensuppgifter som tillhör **Visma Administration 500/1000/2000** respektive **Visma Förening**. Dessa måste vara korrekta för att en koppling ska kunna upprättas. Det sker även en kontroll på vilken version av **Visma Administration 500/1000/2000** respektive **Visma Förening** som är installerad, denna version måste stämma överens med versionen av API:et. Det sker också en kontroll av att företaget eller föreningen är skapat med korrekt version av **Visma Administration** respektive **Visma Förening**.

API:et öppnar den önskade databasen på samma sätt som **Visma Administration 500/1000/2000** respektive **Visma Förening**, därför gäller samma regler för delade data som vid nätverksanvändning av **Visma Administration 500/1000/2000** respektive **Visma Förening**. `AdkOpen()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

Fr.o.m. version 2017.1 av programmet används SQL som databas och företagen där behörighetskontroll är aktiverad använder sig av SQL-servers inloggning. Därför kommer det att krävas inloggning även via Apiet för dessa företag. Observera att detta gäller endast själva inloggningen. Apiet använder sig inte av övriga behörighetskontroller som programmet.

För bakåtkompatibilitet med integrationer gjorda för äldre versioner har följande mekanismer införts:

- Inloggning med hjälp av miljövariabler

- Via programmet lagra integrationsanvändare i registret
- Inloggningsdialog i Apiet med lagring i minnet samt windows register

Företag utan behörighet aktiverad fungerar i princip som förut och omfattas inte av dessa mekanismer.

AdkOpen () har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Dock är AdkNet4Wrapper.dll bakåtkompatibel och funktionen kommer fortsätta att fungera som innan.

Miljövaribler

ADK_USERNAME och ADK_PASSWORD sätts i klartext och är de inloggningsuppgifter användaren ger vid inloggning i företaget vid öppnande av företag i programmet. Dessa miljövaribler används endast om apiet används i en miljö utan användargränssnitt, dvs i en service. Har dessa använts och är felaktiga returneras felkod vid inloggningsförsök och anropet till `AdkOpen ()` misslyckas. Dessa miljövaribler är inte knutna till specifik företagsdatabas, och således behöver denna användare läggas upp i flera företagsdatabaser vid behov.

Lagra integrationsanvändare i programmet

Användare för integration kan lagras med hjälp av programmet. Är programkörningen inom en service har miljövaribler enligt ovan företräde mot denna lagring av användare. Inställningen finns i fönstret för Användare (Arkiv/Användare) och nås genom högerklick i tabellen. Dialog för att mata in markerad användares lösenord visas. Detta behövs för verifiering, så att felaktigt lösenord inte lagras. Den användare som är markerad i tabellen kommer därefter lagras i registret med lösenord (båda uppgifterna är krypterade) under HKEY_LOCAL_MACHINE och gäller således för alla windowsanvändare på aktuell dator. Det finns också ett menyalternativ vid högerklick för att radera användaren i registret. Inställningen lagras per företagsdatabas, så flera inloggningar kan lagras.

Menyn är bara synlig om licens för integration är aktiverad, samt om behörighetskontroll är aktiverad för företaget.

Att ändra i denna plats i registret kräver administratörsrättigheter.

Det går att lagra en användare som ska användas för specificerat företag via kommandoraden. Eleverad prompt är att föredra (eleveringsförsök genomförs annars automatiskt). Anropet har följande syntax:

```
SpcsAdm.exe /apicred /c DBNAME /u SIGN /p PWD
```

DBNAME – databasens namn i SQL-servern (kan hittas i ftg.ini under [DataSource] DBName=).

SIGN – signatur enl medarbetarregistret i företaget

PWD – medarbetarens lösenord

Vid detta anrop verifieras inte lösenordet.

Det går även att radera lagrad användare för specificerat företag med syntaxen:

```
SpcsAdm.exe /apicred /c DBNAME /r
```

Inloggningsdialog i Apiet

Om inget av ovan finns och inloggning krävs i företaget, så kommer Apiet att visa en inloggningsdialog. Dialogen visas inte när Apiet används inuti en service, då dessa inte har ett användargränssnitt. Vid korrekt inloggning sparas användarnamn och lösenord i minnet under programkörningen för att användaren ska slippa ange dessa uppgifter vid varje anrop till `AdkOpen ()`. Användaren bereds möjlighet att spara användarnamn och lösenord på datorn och det lagras då i registret under HKEY_CURRENT_USER. Inställningen gäller således för aktuell företagsdatabas på aktuell datorn för aktuell windowsanvändare. Har användare valt att spara lösenordet på datorn undviks inloggning vid efterföljande programkörningar.

AdkOpen2

C++

```
ADK_ERROR AdkOpen2(CHAR* pszSystemPath, CHAR* pszFtgPath, CHAR*  
pszUserName, CHAR* pszPassword)
```

C#

```
ADKERROR AdkOpen2(string pszSystemPath, string pszFtgPath, string  
pszUserName, string pszPassword)
```

Information

Funktionen är en vidareutveckling av `AdkOpen()` som även tar användarnamn och lösenord som inparametrar. `AdkOpen2()` fungerar som `AdkOpen()`, med undantag från de mekanismer för bakåtkompatibilitet som lagts till i den.

Vid felaktiga inloggningsuppgifter kommer anropet att misslyckas på företag med behörighet aktiverad i programmet. Är behörighet inte aktiverad på företagsdatabasen ignoreras angivet användarnamn och lösenord.

Användarnamn och lösenord angivna i anrop mot `AdkOpen2()` kommer inte lagras i minnet eller på datorn, så det är upp till anropande integrationslösning att hantera dessa uppgifter.

För att ta reda på om användarnamn och lösenord krävs kan funktionen `AdkIsLoginActivated()` användas.

AdkOpen2() finns i AdkNet4Wrapper i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkOpenEx

C++

```
ADK_ERROR AdkOpenEx(CHAR* pszSystemPath, CHAR* pszFtgPath, long lLogOn)
```

Information

Funktionen `AdkOpenEx()` används för att öppna en databas till ett befintligt företag eller förening. Funktionen tar tre inparametrar, den första är en sträng som pekar ut var **Visma Administration 500/1000/2000** respektive **Visma Förening** är installerat och den andra är en sträng som pekar ut var önskat företag eller förening finns. Den tredje används för att slå av respektive på loggning av funktionsanrop via `adk.dll`. Värdet 0 slår av loggning, värdet 1 slår på loggning.

Resultatet av loggningen sparas i fil `APIAdkLogg.txt` som finns i katalogen för gemensamma filer i installationen av **Visma Administration 500/1000/2000** respektive **Visma Förening**.

I övrigt fungerar `AdkOpenEx()` på samma sätt som `AdkOpen()`. Detta gäller även mekanismerna för bakåtkompatibilitet.

AdkOpenEx () har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Dock är AdkNet4Wrapper.dll bakåtkompatibel och funktionen kommer fortsätta att fungera som innan.

AdkOpenEx2

C++

```
ADK_ERROR AdkOpenEx2 (CHAR* pszSystemPath, CHAR* pszFtgPath, CHAR*  
pszUserName, CHAR* pszPassword, long lLogOn)
```

C#

```
ADKERROR AdkOpenEx2 (string pszSystemPath, string pszFtgPath, string  
pszUserName, string pszPassword, int lLogOn)
```

Information

Funktionen är en vidareutveckling av AdkOpenEx () som även tar användarnamn och lösenord som inparametrar. AdkOpenEx2 () fungera som AdkOpenEx (), med undantag från de mekanismer för bakåtkompatibilitet som lagts till i den.

Vid felaktiga inloggningsuppgifter kommer anropet att misslyckas på företag med behörighet aktiverad i programmet.

Användarnamn och lösenord angivna i anrop mot AdkOpenEx2 () kommer inte lagras i minnet eller på datorn, så det är upp till anropande integrationslösning att hantera dessa uppgifter.

AdkOpenEx2 () finns i AdkNet4Wrapper i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkIsLoginActivated

C++

```
ADK_ERROR AdkIsLoginActivated (CHAR* pszSystemPath, CHAR* pszFtgPath,  
LONG* piValue)
```

C#

```
ADKERROR AdkIsLoginActivated (string pszSystemPath, string pszFtgPath, out  
bool pbValue)
```

Information

Funktionen tar reda på om behörighetskontrollen är aktiverad i aktuellt företag. Funktionen kan köras på företagsdatabaser utan att AdkOpen () körts innan. Funktionen kräver inte heller licens för integration.

Svaret lagras i minnet som pekars ut av `piValue`. Värde 0 indikerar att behörighet inte aktiverats. Värde 1 indikerar att behörighet har aktiverats.

Eventuella fel vid anropet returneras i `ADK_ERROR`-strukturen, och värdet i `piValue` är inte pålitligt om `lRc`-variabeln avviker från `ADKE_OK` (dvs 0).

`AdkIsLoginActivated()` finns i `AdkNet4Wrapper` i två varianter. Den äldre har deklarerats som `deprecated` i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkClose

C++

```
VOID AdkClose()
```

C#

```
int AdkClose();
```

Funktionen `AdkClose()` returnerar i .NET en `int`. I nuvarande version kommer alltid 0 att returneras.

Information

Funktionen `AdkClose()` stänger den koppling mot databasen som API:et har öppnat.

AdkCreateData

C++

```
PADK_DATA AdkCreateData(INT iDatabaseId)
```

C#

```
int AdkCreateData(int IDataBaseId)
```

Information

Funktionen `AdkCreateData()` skapar en datastruktur. `AdkCreateData()` tar en inparameter, som anger vilken typ av `ADK_DATA`-struktur som ska skapas. `AdkCreateData()` allokerar minne och initierar en datastruktur av önskad typ. I funktionen `AdkCreateData()` skapas de fält som tillhör den önskade datastrukturen och dessa fält tilldelas sina egenskaper. Även information om datastrukturen, `metaData`, kommer att skapas.

Det minne som allokerats av `AdkCreateData()` ska avallokeras med funktionen `AdkDeleteStruct()`. Observera att datastrukturer för arbete mot databasen endast får skapas av `AdkCreateData()` eller `AdkCreateDataRow()` och endast tas bort med `AdkDeleteStruct()`. Undantaget är funktionen `AdkDelete()` som även tar bort aktuell datastruktur. `AdkCreateData()` returnerar en pekare till den skapade datastrukturen.

AdkCreateDataRow

C++

```
PADK_DATA AdkCreateDataRow(INT lDatabaseId, INT iNumber)
```

C#

```
int AdkCreateDataRow(int IDatabaseId, int iNumber)
```

Information

Funktionen `AdkCreateDataRow()` skapar en datastruktur. `AdkCreateDataRow()` tar två inparametrar, som anger vilken typ av och antalet `ADK_DATA`-strukturer som ska skapas. `AdkCreateDataRow()` fungerar på liknande sätt som `AdkCreateData()` men med skillnaden att det är möjligt att skapa en vektor av datastrukturer. Hur många element som ska ingå i vektorn bestäms med inparameter nummer två.

`AdkCreateDataRow()` används vid skapandet av order-, offert-, faktura- eller leverantörsfaktura-rader. I dessa fall är det ofta önskvärt att skapa flera rader som ska kopplas till ett huvuddokument. `AdkCreateDataRow()` skapar ett antal element med information om dessa, `metaData`, vilken kommer att vara likadan för samtliga element i den skapade vektorn. Alla element i den skapade vektorn kommer att ha lika många aktiverade fält och de kommer att ha samma egenskaper. För att kunna peka ut ett element och arbeta med det, finns funktionen `AdkGetDataRow()` som kommer att beskrivas senare.

Observera att det minne som allokerats med `AdkCreateDataRow()` inte behöver avallokeras. Detta minne avallokeras automatiskt när minnet till det huvuddokument vilket raderna tillhör avallokeras. `AdkCreateDataRow()` returnerar en pekare till den skapade datastrukturen.

AdkDeleteStruct

C++

```
ADK_ERROR AdkDeleteStruct(ADK_DATA* pData)
```

C#

```
ADKERROR AdkDeleteStruct(int pDb)
```

Information

Funktionen `AdkDeleteStruct()` avallokerar det minne som allokerats till en datastruktur skapad med funktionen `AdkCreateData()` samt tillhörande datastrukturer skapade med funktionen `AdkCreateDataRow()`. `AdkDeleteStruct()` tar en inparameter, pekaren till den datastruktur som ska avallokeras. Om t ex en datastruktur av typen "Faktura" är skapad och denna innehåller fem påkopplade rader behöver endast `AdkDeleteStruct()` anropas med fakturahuvudet som inparameter. När detta sker kommer även de tillhörande radernas allokerade minne att avallokeras. Funktionen `AdkDeleteStruct()` bör alltid anropas då arbetet med en datastruktur är klart. `AdkDeleteStruct()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

OBS! `pData` sätts inte till `NULL` av `AdkDeleteStruct()`. Kontroll av hur anropet till `AdkDeleteStruct()` utföll, kan därför inte göras med ledning av värdet av `pData`.

AdkResetStruct

C++

```
ADK_ERROR AdkResetStruct(ADK_DATA* pData)
```

C#

```
ADKERROR AdkResetStruct(int pDb)
```

Information

Funktionen `AdkResetStruct()` nollställer det minne som allokerats till en datastruktur skapad med funktionen `AdkCreateData()` samt avallokerar tillhörande datastrukturer skapade med funktionen `AdkCreateDataRow()`. `AdkResetStruct()` tar en inparameter, pekaren till den datastruktur som ska nollställas. Om t ex en datastruktur av typen "Faktura" är skapad och denna innehåller fem påkopplade rader behöver endast `AdkResetStruct()` anropas med datastrukturen som inparameter. När detta sker kommer de tillhörande radernas allokerade minne att avallokeras. Funktionen `AdkResetStruct()` kan anropas då behovet av nollställning av den allokerade datastrukturen finns. `AdkResetStruct()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

ADK_ERROR

Deklaration

Här följer en deklARATION av datastrukturen.

```
Typedef struct _Error
{
    LONG lRc;
    LONG lDbTable;
    LONG lField;
    LONG lFunction;
    LONG lProgramPart;
} ADK_ERROR, *PADK_ERROR;
```

Information

För felhantering används en datastruktur av typen `ADK_ERROR`. Denna datastruktur innehåller ett antal koder som har olika uppgifter. De koder som finns i datastrukturen är alla av datatypen `long`. Varje kod är till för att beskriva en specifik del i ett felmeddelande och tillsammans bildar de ett felmeddelande. Värdet 0 innebär att allt har gått bra och inget fel har uppstått. Nedan kommer de fem koder som ingår i `ADK_ERROR` att beskrivas:

- `lRc` – Den viktigaste koden eftersom den innehåller själva felmeddelandet som returneras. Är `lRc` lika med 0 har det alltid gått bra, är den skild från 0 har det alltid gått fel. Alla valideringar av utförandet av funktioner ska ske mot detta värde.
- `lDbTable` – Koden för den databastabell som användes när felet uppstod. `lDbTable` behöver inte alltid anta ett värde som skiljer sig från 0 trots att ett fel har uppstått. Observera att dessa värden för databastabeller är definierade i `AdkError.h` bland felkoderna, och att dessa inte överensstämmer med tabelldefinitionerna i `Adk.h`

- `lField` – Koderna för de fält i aktuell databas som användes när felet uppstod. `lField` behöver inte alltid anta ett värde som skiljer sig från 0 trots att ett fel har uppstått. Detta värde överensstämmer med värdena definierade för fält i `Adk.h`.
- `lFunction` – Koderna för den funktion som anropades när felet uppstod.
- `lProgramPart` – Koderna för den del i API:et där felet uppstod.

Om felkoden som returneras i `lRc` har värdet 1000 eller högre, så har felet uppstått i grundstrukturen.

AdkGetErrorText

C++

```
VOID AdkGetErrorText(ADK_ERROR* error, ADK_ERROR_TEXT_TYPE
nErrorTextType, CHAR** achBuf, INT iLen)
```

C#

```
string AdkGetErrorText(ADKERROR Error_Renamed, ADK_ERROR_TEXT_TYPE
nErrorTextType)
```

Information

Till alla de felkoder som kan uppkomma finns texter som kan hämtas med hjälp av funktionen `AdkGetErrorText()`. Det är endast möjligt att hämta en text i taget. Det kan finnas flera olika felkoder i ett felmeddelande och därför kan `AdkGetErrorText()` behöva anropas upp till fem gånger för att få samtliga texter som tillhör aktuellt fel. `AdkGetErrorText()` tar fyra inparametrar:

- `ADK_ERROR* error` – En pekare till datastrukturen med felmeddelanden.
- `ADK_ERROR_TEXT_TYPE nErrorTextType` – Vilken typ av fel som ska hämtas. De typer av fel som finns motsvarar de koder som finns i datastrukturen `ADK_ERROR` och är definierade i `ADK_ERROR_TEXT_TYPE` som finns definierad i `Adk.h`.
- `CHAR** achBuf` – Minnesadressen till en `char`-pekare som är definierad utanför API:et. Här kommer feltexten att skrivas. *(Ej relevant i .NET)*
- `INT iLen` – `Int` som definierar hur många tecken som finns i `achBuf`. *(Ej relevant i .NET)*

För ytterligare information om felkoder hänvisas till bilaga ***Beskrivning av felkoder***.

Funktionen `AdkGetErrorText()` returnerar i .NET en sträng med sökt feltext istället för via inparametrar.

AdkSetFilter

C++

```
ADK_ERROR AdkSetFilter(ADK_DATA* pData, int iFieldId, const char*
pExpression, int iVersion)
```

C#

```
ADKERROR AdkSetFilter(int pDb, int iFieldId, string pExpression, int
iVersion)
```

Information

`AdkSetFilter` används för att sätta filter på ett fält i en `pData`-struktur. Filtrering innebär att bara poster som uppfyller villkoret i filtret kan nås med `AdFirstEx/AdkNextEx/AdkLastEx/AdkPreviousEx/AdkFind2` och dess varianter. Filter kan sättas på ett eller flera fält. Om filter sätts på fler fält träffas bara poster som uppfyller villkoret i alla filter. Med filtrering kan man minska antal poster att traversera och därigenom utföra vissa uppgifter effektivare. Funktionen `AdkSetFilter` tar fyra inparametrar:

- `ADK_DATA* pData` – En pekare till den datastruktur filter ska sättas på.
- `int iFieldId` – Fält filtrering önskas på.
- `const char* pExpression` – Uttryck för filtrering. Uttryckets syntax är detsamma som sök/filtrering i programmet. Finns ett filter sedan innan på aktuellt fält kommer det skrivas över av det nya om det är ett godkänt uttryck. Tom sträng nollställer filtret för aktuellt fält.
- `int iVersion` – Version av filtreringsfunktionen. I denna version ska den alltid ha värdet 0.

`AdkSetFilter` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Det finns några specifika felkoder för filtreringsfunktioner, som specificeras nedan. Se avsnittet `ADK_ERROR` för ytterligare information om övriga felkoder.

`AdkSetFilter()` finns i `AdkNet4Wrapper` i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

Filtersyntax

```
NOT -> '!' | 'INTE'
OR -> ',' | 'ELLER'
AND -> 'OCH'
BETWEEN -> '..'
```

```
OP -> '>' | '<' | '<>' | '>=' | '<=' | '='
```

```
TERM -> EXPRTEXT | OP EXPRTEXT | EXPRTEXT BETWEEN EXPRTEXT | '('EXPR')'
```

```
EXPR -> NOT EXPR | TERM AND EXPR | TERM OR EXPR | TERM
```

Operatorer:

Operator	Ställighet	Innebörd	Exempel
OCH	Binär (tvåställig)	Och	<100 OCH >200
ELLER	Binär	Eller	100 ELLER 200 100,200
INTE !	Unär (ettställig)	Inte	INTE 100 !100
<	Unär	Mindre än	<100
>	Unär	Större än	>100
<=	Unär	Mindre än eller lika med	<=100
>=	Unär	Större än eller lika med	>=100
<>	Unär	Skilt från	<>100
=	Unär	Resten av uttrycket är en operand	=100
..	Binär	Intervall	100..200

Övriga reserverade tecken:

Tecken	Fälttyp	Innebörd
*	Text	Vilka tecken som helst (0-n st)
?	Text	Vilket tecken som helst (1 st)
.	Numeriska	Decimalavskiljare
Ja J 1	Logiska	Sant
Nej N 0	Logiska	Falskt

Filtersträngar är inte skiftlägeskänsliga (case sensitive).

Observera att i strängfält filtreras på exakt sträng. Filtreras man på "stor" så ger det inte träff på tex "storstad". Detta är en skillnad dels mot programmets filtrering i tabeller där sträng utan ytterligare operatorer ger träff på inledande sträng, dels skillnad mot hur AdkFind2 fungerar.

Om man vill filtrera på exakta värden, och dessa tex byggs ihop i en egen funktion, är det att rekommendera att inleda söksträngen med operatörn =, så kommer sökvärdet inte att tolkas vidare av API:et.

Vid ändring av filter, med hjälp av `AdkSetFilter` kommer alltid en reset göras av datastrukturen. Det är därför rekommenderat att antingen köra `AdkFirstEx` eller `AdkLastEx` eller `AdkFind2` efter att man satt sina filter. `AdkNextEx` och `AdkPreviousEx` har ett odefinierat beteende direkt efter `AdkSetFilter` körts. (Detta gäller alla varianter av nämnda funktioner).

Specifika felkoder

`ADKE_INVALID_FILTER_FIELD` - angivet fält kan inte filtreras

`ADKE_INVALID_FILTER_EXPRESSION` - ogiltigt uttryck (om logg är aktiverad kommer mer specifik feltext visas `APIAdkLogg.txt`).

`ADKE_INVALID_FILTER_SYNTAX_VERSION` - angiven version stöds inte (0 för 2018.6)

AdkIsFilterable

C++

```
ADK_ERROR AdkIsFilterable(ADK_DATA* pData, INT iFieldId, BOOL*
pbIsFilterable)
```

C#

```
ADKERROR AdkIsFilterable(int pDb, int iFieldId, out bool pbIsFilterable)
```

Information

`AdkIsFilterable` används för att ta reda på om man kan sätta filter på ett fält i en `pData`-struktur. Funktionen `AdkIsFilterable` tar tre inparametrar:

- `ADK_DATA* pData` – En pekare till den datastruktur filter ska sättas på.
- `int iFieldId` – Fält filtrering önskas på.
- `BOOL* pbIsFilterable` – Utparameter. Svarar på om fältet är filtrerbart. Om variabeln `IRc` i returnerad error-struktur är lika med 0 (`ADKE_OK`) så betyder värdet 1 (`TRUE`) i `pbIsFilterable` att fältet är filtrerbart och värdet 0 (`FALSE`) att fältet inte är filtrerbart.

`AdkIsFilterable` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information om felkoder.

`AdkIsFilterable()` finns i `AdkNet4Wrapper` i två varianter. Den äldre har deklarerats som `deprecated` i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkClearFilters

C++

```
ADK_ERROR AdkClearFilters(ADK_DATA* pData)
```

C#

```
ADKERROR AdkClearFilters(int pDb)
```

Information

`AdkClearFilters` används för att rensa alla filter i en `pData`-struktur. Funktionen `AdkClearFilters` tar en inparameter:

- `ADK_DATA* pData` – En pekare till den datastruktur filter ska sättas på.

`AdkSetFilter` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information om felkoder.

Vid ändring av filter, med hjälp av `AdkClearFilters` kommer alltid den underliggande hanteraren att gå till den första posten i sökordningen - men datastrukturen laddas inte med fräska data. Det är därför rekommenderat att antingen köra `AdkFirstEx` eller `AdkLastEx` eller `AdkFind2` efter att man satt sina filter. `AdkNextEx` och `AdkPreviousEx` har ett odefinierat beteende direkt efter `AdkSetFilter` körts. (Detta gäller alla varianter av nämnda funktioner).

AdkLongToDate och AdkDateToLong

Information

I databasen behandlas datum som datatypen `long` (Juliandatum). För att tilldela eller läsa ett datum ska funktionerna `AdkSetDate()` respektive `AdkGetDate()` användas. För att underlätta hanteringen av datum tillhandahåller API:et två datumfunktioner för konvertering till och från Juliandatum, `AdkLongToDate()` och `AdkDateToLong()`.

AdkLongToDate() och AdkDateToLong() har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionerna är inte relevanta då det nu finns set- och get-gunktioner anpassade till datatyperna i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och funktionen kommer fortsätta att fungera som innan.

AdkLongToDate

C++

```
ADK_ERROR AdkLongToDate(LONG lValue, CHAR** ppsValue, INT iLen)
```

Information

`AdkLongToDate` används när ett datum i formen av `long` (Juliandatum) erhållits och ska konverteras till ett datum i form av en sträng. När t ex ett datum hämtas från databasen returneras det som en `long` (Juliandatum). För att få datumet i form av en sträng används funktionen `AdkLongToDate` som returnerar en sträng i formatet "YYYY-MM-DD", är datorns datumformat annat kommer strängen att returneras i detta format. Det finns även fält som är så kallade tidstämplar, det är de datumfält som innehåller en 10 tecken lång `long` istället för de normala 8. När dessa datum hämtas returneras en sträng i formatet "YYYY-MM-DD HH:mm:ss". Funktionen `AdkLongToDate` tar tre inparametrar:

- `LONG lValue` – Det värde som ska omvandlas. Är detta värde felaktigt kommer dagens datum att returneras.
- `CHAR** ppsValue` – Minnesadressen till en `charpekare` som är definierad utanför API:et. Här kommer datumet att skrivas i form av en sträng. Denna `charpekare` måste innehålla minst 11 tecken. Är fältet en så kallad tidstämpel så ska strängen vara minst 20 tecken om hela stämpeln önskas. Är strängen mindre än 20 tecken kommer datumet returneras enligt "YYYY-MM-DD", är datorns datumformat annat kommer strängen att returneras i detta format.
- `INT iLen` – Heltal som definierar hur många tecken som finns i `ppsValue`.

`AdkLongToDate` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkDateToLong

C++

```
ADK_ERROR AdkDateToLong(CHAR* sValue, LONG* plValue)
```

Information

`AdkDateToLong` används när ett datum finns i en sträng och ska konverteras till en `long` (Juliandatum) för att kunna skrivas till databasen. Funktionen `AdkDateToLong` tar två inparametrar:

- `CHAR* sValue` – En charpekare som innehåller det datum som ska omvandlas. Formen på datum är normalt "YYYY-MM-DD" på en dator med svenska inställningar. Är datumformatet annat ska detta användas. Kan funktionen inte tolka datumsträngen kommer värdet för dagens datum att erhållas.
- `LONG* plValue` – Minnesadressen till en `long` där det konverterade värdet kommer att skrivas.

`AdkLongToDate` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkLongToDateTime och AdkDateTimeToLong

Information

I databasen behandlas tidsstämplar som datatypen `long (__time32_t)`. För att tilldela eller läsa en tidsstämpel ska funktionerna `AdkSetDate()` respektive `AdkGetDate()` användas. För att underlätta hanteringen av tidsstämplar tillhandahåller API:et två funktioner för konvertering till och från `__time32_t`, `AdkLongToDateTime()` och `AdkDateTimeToLong()`.

`AdkLongToDateTime()` och `AdkDateTimeToLong()` har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionerna är inte relevanta då det nu finns set- och get-gunktioner anpassade till datatyperna i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och funktionen kommer fortsätta att fungera som innan.

AdkLongToDateTime

C++

```
ADK_ERROR AdkLongToDateTime(LONG lValue, CHAR** ppsValue, INT iLen)
```

Information

`AdkLongToDateTime` används när tidsstämpel i formen av `long (__time32_t)` erhållits och ska konverteras till ett datum i form av en sträng. När t ex en tidsstämpel hämtas från databasen returneras det som en `long (__time32_t)`. För att få tidsstämpeln i form av en sträng används funktionen `AdkLongToDateTime` som returnerar en sträng i formatet "YYYY-MM-DD HH:mm:ss", strängen returneras alltid i detta format. Funktionen `AdkLongToDateTime` tar tre inparametrar:

- `LONG lValue` – Det värde som ska omvandlas.
- `CHAR** ppsValue` – Minnesadressen till en charpekare som är definierad utanför API:et. Här kommer tidsstämpeln att skrivas i form av en sträng. Denna charpekare måste innehålla minst 20 tecken.
- `INT iLen` – Heltal som definierar hur många tecken som finns i `ppsValue`.

`AdkLongToDateTime` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

`AdkLongToDate` kan också användas för att omvandla en tidsstämpel till en sträng (se avsnittet ovan).

AdkDateTimeToLong

C++

```
ADK_ERROR AdkDateTimeToLong (CHAR* sValue, LONG* plValue)
```

Information

`AdkDateToLong` används när tidsstämpel finns i en sträng och ska konverteras till en `long` (`__time32_t`) för att kunna skrivas till databasen. Funktionen `AdkDateTimeToLong` tar två inparametrar:

- `CHAR* sValue` – En charpekare som innehåller den tidsstämpel som ska omvandlas. Formen på tidsstämpeln ska vara "YYYY-MM-DD HH:mm:ss". Kan funktionen inte tolka datumsträngen kommer felkoden `ADKE_INVALID_DATETIME_STRING` att erhållas.
- `LONG* plValue` – Minnesadressen till en `long` där det konverterade värdet kommer att skrivas.

`AdkLongToDate` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkGet

Information

`AdkGet()` är en samling funktioner som hämtar data ur en datastruktur. `AdkGet()` används för att läsa data i en datastruktur efter att en post hämtats i databasen med någon av funktionerna `AdkFind()`, `AdkFirst()`, `AdkLast()`, `AdkPrevious()` eller `AdkNext()`. Vilken av de olika `AdkGet`-funktionerna som ska används beror på vilken datatyp det fält har där ett värde ska hämtas.

AdkGetStr

C++

```
ADK_ERROR AdkGetStr (ADK_DATA* pData, INT iFieldId, CHAR** ppsValue, INT iLen)
```

C#

```
ADKERROR AdkGetStr (int pDb, int iFieldId, out string ppsValue)
```

Information

`AdkGetStr()` hämtar en sträng från ett fält i en angiven datastruktur. Det fält där ett värde ska hämtas måste vara av typen `eChar`. Fältets egenskap `iType` ska ha värdet `eChar`. Funktionen `AdkGetStr()` tar fyra inparametrar:

- `ADK_DATA* pData` – En pekare till den datastruktur där ett värde ska hämtas.
- `INT iFieldId` – Id till det fält där ett värde ska hämtas.
- `CHAR** ppsValue` – Minnesadressen till en `char`-pekare som är definierad utanför API:et. Värdet kommer att skrivas in där.
- `INT iLen` – Heltal som definierar hur många tecken som finns i `ppsValue`.

`AdkGetStr()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

`AdkGetStr()` finns i `AdkNet4Wrapper` i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetBool

C++

```
ADK_ERROR AdkGetBool(ADK_DATA* pData, INT iFieldId, INT* piValue)
```

C#

```
ADKERROR AdkGetBool(int pDb, int iFieldId, out bool pbValue)
```

Information

`AdkGetBool()` hämtar en `BOOL` från ett fält i en angiven datastruktur. Det fält där ett värde ska hämtas måste vara av typen `eBool`. Fältets egenskap `iType` ska ha värdet `eBool`. `AdkGetBool()` tar tre inparametrar:

- `ADK_DATA* pData` – En pekare till den datastrukturen där ett värde ska hämtas.
- `INT iFieldId` – Id till det fält där ett värde ska hämtas.
- `INT* piValue` – Minnesadressen till en `int` som är definierad utanför API:et. Värdet kommer att skrivas in där. En `BOOL` i API:et behandlas som en `int`, vilket gör att en `int`-pekare kan skickas till denna parameter.

`AdkGetBool()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

`AdkGetBool()` finns i `AdkNet4Wrapper` i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetData (AdkGetRowData)

C++

```
ADK_ERROR AdkGetData(ADK_DATA* pData, INT iFieldId, INT iIx, ADK_DATA** ppDataRow)
```

C#

```
ADKERROR AdkGetRowData(int pParentDb, int iIndex, out int pDbRow)
```

Information

`AdkGetData()` hämtar en `ADK_DATA`-pekare från ett fält i en angiven datastruktur. Det fält där ett värde ska hämtas måste vara av typen `eData`. Fältets egenskap `iType` ska ha värdet `eData`. `AdkGetData()` används när pekaren till en rad som tillhör en order, offert, faktura eller leverantörsfaktura ska hämtas. Som tidigare behandlats kan en datastruktur med rader innehålla flera element och därmed vara en vektor. Det är dock endast möjligt att hämta pekaren till en rad i en sådan vektor. Vilken rad som önskas, anges med ett index. `AdkGetData()`. Funktionen tar fyra inparametrar:

- `ADK_DATA* pData` – En pekare till den datastruktur där ett värde ska hämtas.
- `INT iFieldId` – Id till det fält där ett värde ska hämtas.
- `INT iIx` – Index till det element som ska hämtas.
- `ADK_DATA** ppDataRow` – Minnesadressen till en `ADK_DATA`-pekare som är definierad utanför API:et. Värdet kommer att skrivas in där.

`AdkGetData()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkGetDouble (AdkGetNumeric)

C++

```
ADK_ERROR AdkGetDouble(ADK_DATA* pData, INT iFieldId, DOUBLE* pdValue)
```

C#

```
ADKERROR AdkGetNumeric(int pDb, int iFieldId, out double pdValue)  
ADKERROR AdkGetNumeric(int pDb, int iFieldId, out decimal pdValue)
```

Information

`AdkGetDouble()` hämtar en `double` från ett fält i en angiven datastruktur. Det fält där ett värde ska hämtas måste vara av typen `edouble`. Fältets egenskap `iType` ska ha värdet `eDouble`. `AdkGetDouble()` tar tre inparametrar:

- `ADK_DATA* pData` – En pekare till den datastruktur där ett värde ska hämtas.
- `INT iFieldId` – Id till det fält där ett värde ska hämtas.

- `DOUBLE* pdValue` – Minnesadressen till en `double` som är definierad utanför API:et. Värdet kommer att skrivas in där.

`AdkGetDouble()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

`AdkGetDouble()` har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då `AdkGetNumeric()` med dess två varianter ersätter funktionen i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och funktionen kommer fortsätta att fungera som innan.

AdkGetDate

C++

```
ADK_ERROR AdkGetDate(ADK_DATA* pData, INT iFieldId, LONG* plValue)
```

C#

```
ADKERROR AdkGetDate(int pDb, int iFieldId, out DateTime? pdaValue)
```

Information

`AdkGetDate()` hämtar en `date` från ett fält i en angiven datastruktur. Det fält där ett värde ska hämtas måste vara av typen `eDate`. Fältets egenskap `iType` ska ha värdet `eDate`. `AdkGetDate()` tar tre inparametrar:

- `ADK_DATA* pData` – En pekare till den datastruktur där ett värde ska hämtas.
- `INT iFieldId` – Id till det fält där ett värde ska hämtas.
- `LONG* plValue` – Minnesadressen till en `long` som är definierad utanför API:et. Värdet kommer att skrivas in där. Ett datum i API:et behandlas som en `long`, vilket gör att det är en `long` som ska skickas till denna parameter. För att konvertera datum från `long` (Juliandatum) till en sträng kan funktionen `AdkLongToDate` anropas.

`AdkGetDate()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

`AdkGetDate()` finns i `AdkNet4Wrapper` i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkSet

Information

AdkSet är ett antal funktioner som skriver värden till en datastruktur. AdkSet används innan en post ska skrivas till databasen med någon av funktionerna AdkAdd() eller AdkUpdate(). Vilken av de olika AdkSet-funktionerna som ska används beror på vilken datatyp fältet har, till vilket ett värde ska skrivas.

AdkSetStr

C++

```
ADK_ERROR AdkSetStr(ADK_DATA* pData, INT iFieldId, CHAR* pValue)
```

C#

```
ADKERROR AdkSetStr(int pDb, int iFieldId, string pValue)
```

Information

AdkSetStr() skriver en sträng till ett fält i en angiven datastruktur. Det fält där ett värde ska tilldelas måste vara av typen eChar, fältets egenskap iType ska ha värdet eChar. Funktionen AdkSetStr() tar tre inparametrar:

- ADK_DATA* pData – En pekare till den datastruktur där ett värde ska tilldelas.
- INT iFieldId – Id till det fält där ett värde ska tilldelas.
- CHAR* pValue – En char-pekare som innehåller det värde som ska tilldelas till aktuellt fält.

AdkSetStr() returnerar en datastruktur av typen ADK_ERROR, vilken innehåller information om resultatet av funktionen. Se avsnittet ADK_ERROR för ytterligare information.

Ett specialfall finns, och det är när radering av innehållet i ett textfält önskas. I det läget ska en sträng med ett mellanslag (" ") skickas in i anropet. Detta tolkas av funktionen som radering av innehållet i fältet önskas.

AdkSetStr() finns i AdkNet4Wrapper i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkSetBool

C++

```
ADK_ERROR AdkSetBool(ADK_DATA* pData, INT iFieldId, INT iValue)
```

C#

```
ADKERROR AdkSetBool(int pDb, int iFieldId, bool bValue)
```

Information

`AdkSetBool()` skriver en `BOOL` till ett fält i en angiven datastruktur. Det fält där ett värde ska tilldelas måste vara av typen `eBool`. Fältets egenskap `iType` ska ha värdet `eBool`. `AdkSetBool()` tar tre inparametrar:

- `ADK_DATA* pData` – En pekare till den datastruktur där ett värde ska tilldelas.
- `INT iFieldId` – Id till det fält där ett värde ska tilldelas.
- `INT iValue` – En `int` med det värde som ska tilldelas till aktuellt fält. En `BOOL` i API:et behandlas som en `int`, vilket gör att en `int` kan skickas till denna parameter.

`AdkSetBool()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

`AdkSetBool()` finns i `AdkNet4Wrapper` i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkSetData

C++

```
ADK_ERROR AdkSetData(ADK_DATA* pData, INT iFieldId, ADK_DATA* pDataRow)
```

C#

```
ADKERROR AdkSetData(int pDb, int iFieldId, int pDataRow)
```

Information

`AdkSetData()` tilldelar en `ADK_DATA`-pekare till ett fält i en angiven datastruktur. Det fält där ett värde ska tilldelas måste vara av typen `eData`. Fältets egenskap `iType` ska ha värdet `eData`. Funktionen `AdkSetData()` används när ett dokumenthuvud i form av en order, offert, faktura eller leverantörsfaktura ska knytas samman med tillhörande rader som finns i en enskild datastruktur. `AdkSetData()` tar tre inparametrar:

- `ADK_DATA* pData` – En pekare till datastrukturen/dokumenthuvudet där ett värde ska tilldelas.
- `INT iFieldId` – Id till det fält där ett värde ska tilldelas.
- `ADK_DATA* pDataRow` – En `ADK_DATA`-pekare med det värde som ska tilldelas till aktuellt fält. Denna parameter ska innehålla en pekare till de dokumentrader som ska kopplas till dokumenthuvudet `pData`.

`AdkSetData()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkSetDouble

C++

```
ADK_ERROR AdkSetDouble(ADK_DATA* pData, INT iFieldId, DOUBLE dValue)
```

C#

```
ADKERROR AdkSetDouble(int pDb, int iFieldId, double dValue)
```

Information

`AdkSetDouble()` tilldelar en `double` till ett fält i en angiven datastruktur. Det fält där ett värde ska tilldelas måste vara av typen `eDouble`. Fältets egenskap `iType` ska ha värdet `eDouble`. Funktionen `AdkSetDouble()` tar tre inparametrar:

- `ADK_DATA* pData` – En pekare till den datastrukturen där ett värde ska tilldelas.
- `INT iFieldId` – Id till det fält där ett värde ska tilldelas.
- `DOUBLE dValue` – Det `double` värde som ska tilldelas till aktuellt fält.

`AdkSetDouble()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkSetDate (AdkSetTimestamp)

C++

```
ADK_ERROR AdkSetDate(ADK_DATA* pData, INT iFieldId, LONG lValue)
```

C#

```
ADKERROR AdkSetDate(int pDb, int iFieldId, DateTime? lValue)  
ADKERROR AdkSetTimestamp(int pDb, int iFieldId, DateTime? lValue)
```

Information

`AdkSetDate()` tilldelar en `date` till ett fält i en angiven datastruktur. Det fält där ett värde ska tilldelas måste vara av typen `eDate`. Fältets egenskap `iType` ska ha värdet `eDate`. Funktionen `AdkSetDate()` tar tre inparametrar:

- `ADK_DATA* pData` – En pekare till den datastrukturen där ett värde ska tilldelas.
- `INT iFieldId` – Id till det fält där ett värde ska tilldelas.
- `LONG lValue` – Det `long` (Juliandatum) värde som ska tilldelas till aktuellt fält. En `date` i API:et behandlas som en `long` (Juliandatum), vilket gör att det är en `long` (Juliandatum) som ska skickas till denna parameter. För att konvertera en sträng till en `long` (Juliandatum) kan funktionen `AdkDateToLong()` anropas.

`AdkSetDate()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

`AdkSetDate()` finns i `AdkNet4Wrapper` i två varianter. Den äldre har deklarerats som `deprecated` i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

`AdkSetTimestamp()` finns i `AdkNet4Wrapper`. Den används för att sätta värden i ett tidsstämpelfält då `AdkSetDate()` i `AdkNet4Wrapper` bara hanterar datumfält. Använd `AdkGetTimestampFieldId()` för att hitta tidsstämpelfältet för aktuell tabell.

AdkGetType

C++

```
ADK_ERROR AdkGetType(ADK_DATA* pData, INT iFieldId, ADK_FIELD_TYPE*
eType)
```

C#

```
ADKERROR AdkGetType(int pDb, int iFieldId, ref ADK_FIELD_TYPE pType)
```

Information

Funktionen `AdkGetType()` hämtar ett fälts datatyp. De olika typer ett fält kan anta finns definierade i enumeren `ADK_FIELD_TYPE` som finns definierad i `Adk.h`. Det krävs en deklaration av en `ADK_FIELD_TYPE` för att funktionen ska kunna användas. Denna variabel kommer att tilldelas den typ som aktuellt fält har. `AdkGetType()` tar tre inparametrar:

- `ADK_DATA* pData` – En pekare till den datastruktur där ett fälts typ ska hämtas.
- `INT iFieldId` – Id till det fält där datatypen ska hämtas.
- `ADK_FIELD_TYPE* peType` – Minnesadressen till en variabel av typen `ADK_FIELD_TYPE`. Här kommer typen att anges.

`AdkGetType()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkGetLength

C++

```
ADK_ERROR AdkGetLength(ADK_DATA* pData, INT iFieldId, INT* iLength)
```

C#

```
ADKERROR AdkGetLength(int pDb, int iFieldId, ref int iLength)
```


Information

Funktionen `AdkGetLength()` hämtar fältlängd för aktuellt fält. Till `AdkGetLength()` skickas en `int`-pekare in. Här kommer längden att skrivas. Funktionen `AdkGetLength()` tar tre inparametrar:

- `ADK_DATA* pData` – En pekare till den datastruktur där fältets längd ska hämtas.
- `INT iFieldId` – Id till det fält där fältlängden ska hämtas.
- `INT* iLength` – Minnesadressen till en `int`. Här kommer längden att anges.

`AdkGetLength()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkGetFieldName

C++

```
ADK_ERROR AdkGetFieldName(INT iDataBaseId, INT iFieldId, CHAR** ppsValue,
INT iLen)
```

C#

```
ADKERROR AdkGetFieldName(int IDatabaseId, int iFieldId, out string
ppsValue)
```

Information

Funktionen `AdkGetFieldName()` hämtar namnet på ett fält i databasen, d v s det alias som finns till varje fält i databasen. Aliaset är inte det samma som namnet på definitionen av fältet som finns i `Adk.h`-filen. Funktionen `AdkGetFieldName()` tar fyra inparametrar:

- `INT iDataBaseId` – Id till den databastabell där ett fältnamn ska hämtas.
- `INT iFieldId` – Id till det fält där fältnamnet ska hämtas.
- `CHAR** ppsValue` – Minnesadressen till en `char`-pekare som är definierad utanför API:et. Här kommer namnet att skrivas.
- `INT iLen` – `int` som definierar antal tecken i `ppsValue`.

`AdkGetFieldName()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkGetFieldName() finns i AdkNet4Wrapper i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetDataRow

C++

```
PADK_DATA AdkGetDataRow(ADK_DATA* pDataRow, INT iIx)
```

C#

```
int AdkGetDataRow(int pDb, int iIndex)
```

Information

Funktionen `AdkGetDataRow()` returnerar en pekare till en önskad rad i en datastruktur med flera element. Funktionen används t ex när en specifik rad ska pekas ut i de fall när arbete ska ske i en struktur med flera fakturarader. Funktionen `AdkGetDataRow()` tar två inparametrar:

- `ADK_DATA* pDataRow` – En pekare till den datastruktur där vi ska peka ut en specifik rad.
- `iIx` – Index till önskad rad, d v s vilken rad i datastrukturen som ska hämtas. Indexet måste vara ett godkänt värde för aktuell datastruktur. Index börjar på 0 vilket motsvarar rad 1.

AdkGetDecimals

C++

```
ADK_ERROR AdkGetDecimals(ADK_DATA* pData, INT iFieldId, INT* iDec)
```

C#

```
int AdkGetDecimals(int pDb, int iFieldId)
```

Information

Funktionen `AdkGetDecimals()` hämtar det antal decimaler ett fält kan innehålla. Till `AdkGetDecimals()` skickas en `int`-pekare in. Här kommer antalet decimaler att skrivas. Funktionen `AdkGetDecimals()` tar tre inparametrar:

- `ADK_DATA* pData` – En pekare till den datastruktur där antalet decimaler ska hämtas från ett fält.
- `INT iFieldId` – Id till det fält där antalet decimaler ska hämtas.
- `INT* iDec` – Minnesadressen till en `int`, här kommer antalet decimaler att anges. (Ej relevant i .NET och nya varianten av funktionen)

Det antal decimaler som returneras motsvarar det antal som fältet kan innehålla enligt databasen, det är inte säkert att det är samma antal som gäller vid användningen av fältet. När t ex rader till en faktura skapas är det enheten på raden som bestämmer vilket antal decimaler som är möjligt att använda. `AdkGetDecimals()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Se avsnittet `ADK_ERROR` för ytterligare information.

AdkGetDecimals() finns i *AdkNet4Wrapper* i två varianter. Den äldre har deklarerats som *deprecated* i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är *AdkNet4Wrapper.dll* bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetTimestampField (AdkGetTimeStampFieldId)

C++

```
ADK_ERROR AdkGetTimestampField(ADK_DATA* pData, INT* iFieldId)
```

C#

```
ADKERROR AdkGetTimestampFieldId(int pDb, out int iFieldId)
```

Information

Funktionen *AdkGetTimestampField()* hämtar det fält i tabellen som håller tidsstämpeln. Till *AdkGetTimestampField()* skickas en *int*-pekare in. Här kommer fältnumret att skrivas in. Funktionen *AdkGetTimestampField()* tar två inparametrar:

- *ADK_DATA* pData* – En pekare till den datastruktur där fältet för tidsstämpeln ska hämtas.
- *INT* iFieldId* – Minnesadressen till en *int*, här kommer fältId skrivas.

AdkGetTimestampField() returnerar en datastruktur av typen *ADK_ERROR*, vilken innehåller information om resultatet av funktionen. Se avsnittet *ADK_ERROR* för ytterligare information. Om tabellen ifråga saknar fält för tidsstämpel returneras *ADKE_TIMESTAMP_FIELD_NOT_FOUND*.

AdkGetTimestampField() finns i *AdkNet4Wrapper* i två varianter. Den äldre har deklarerats som *deprecated* i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är *AdkNet4Wrapper.dll* bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetLastFldId (AdkGetLastFieldId)

C++

```
ADK_ERROR AdkGetLastFldId(ADK_DATA* pData, INT* piFieldId)
```

C#

```
ADKERROR AdkGetLastFieldId(int pDb, out int iFieldId)
```

Information

Funktionen tar reda på fältet med högsta fält-Id för aktuell databastabell. Funktionen kräver en pekare till en korrekt initierad *ADK_DATA*-struktur som inparameter (se avsnittet om *AdkCreateData()* för mer information).

Svaret lagras i minnet som pekas ut av *piFieldId*.

Eventuella fel vid anropet returneras i ADK_ERROR-strukturen, och värdet i piFieldId är inte pålitligt om lrc-variabeln avviker från ADKE_OK (dvs 0).

Möjliga felkoder:

ADKE_INVALID_STRUCT_POINTER, ADKE_NULLPOINTER, ADKE_INVALID_DB.

AdkGetLastFldId() har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten *AdkGetLastFieldId()* ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetNrowsFldId (AdkGetNrowsFieldId)

C++

```
ADK_ERROR AdkGetNrowsFldId(ADK_DATA* pData, INT* piFieldId)
```

C#

```
ADKERROR AdkGetNrowsFieldId(int pDb, out int iFieldId)
```

Information

Funktionen tar reda på fält-Id för antal rader tillhörande dokumentet. Funktionen kräver en pekare till en korrekt initierad ADK_DATA-struktur som inparameter (se avsnittet om AdkCreateData() för mer information).

Svaret lagras i minnet som pekas ut av piFieldId.

Eventuella fel vid anropet returneras i ADK_ERROR-strukturen, och värdet i piFieldId är inte pålitligt om lrc-variabeln avviker från ADKE_OK (dvs 0). Anrop på tabell som saknar rader resulterar i felkod ADKE_INVALID_DB.

Möjliga felkoder:

ADKE_INVALID_STRUCT_POINTER, ADKE_NULLPOINTER, ADKE_INVALID_DB.

AdkGetNrowsFldId() har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten *AdkGetNrowsFieldId()* ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetRowsFldId (AdkGetRowsFieldId)

C++

```
ADK_ERROR AdkGetRowsFldId(ADK_DATA* pData, INT* piFieldId)
```

C#

```
ADKERROR AdkGetRowsFieldId(int pDb, out int iFieldId)
```

Information

Funktionen tar reda på fält-Id för raderna tillhörande dokumentet. Funktionen kräver en pekare till en korrekt initierad ADK_DATA-struktur som inparameter (se avsnittet om AdkCreateData() för mer information).

Svaret lagras i minnet som pekars ut av piFieldId.

Eventuella fel vid anropet returneras i ADK_ERROR-strukturen, och värdet i piFieldId är inte pålitligt om lRc-variabeln avviker från ADKE_OK (dvs 0). Anrop på tabell som saknar rader resulterar i felkod ADKE_INVALID_DB.

Möjliga felkoder:

ADKE_INVALID_STRUCT_POINTER, ADKE_NULLPOINTER, ADKE_INVALID_DB.

AdkGetRowsFldId() har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten AdkGetRowsFieldId() ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetRowDbId (AdkGetRowDataId)

C++

```
ADK_ERROR AdkGetRowDbId(ADK_DATA* pData, INT* piRowDbId)
```

C#

```
ADKERROR AdkGetRowDataId(int pDb, out int iRowDbId)
```

Information

Funktionen tar reda på databas-Id för raderna tillhörande dokumentet. Funktionen kräver en pekare till en korrekt initierad ADK_DATA-struktur som inparameter (se avsnittet om AdkCreateData() för mer information).

Svaret lagras i minnet som pekars ut av piRowDbId.

Eventuella fel vid anropet returneras i ADK_ERROR-strukturen, och värdet i piFieldId är inte pålitligt om lRc-variabeln avviker från ADKE_OK (dvs 0). Anrop på tabell som saknar rader resulterar i felkod ADKE_INVALID_DB.

Möjliga felkoder:

ADKE_INVALID_STRUCT_POINTER, ADKE_NULLPOINTER, ADKE_INVALID_DB.

AdkGetRowDbId() har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten AdkGetRowDataId() ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkIsReadWrite

C++

```
BOOL AdkIsReadWrite(ADK_DATA* pData, INT iFieldId)
```

C#

```
bool AdkIsReadWrite(int pDb, int iFieldId)
```

Information

Funktionen `AdkIsReadWrite()` returnerar en `BOOL` som anger om ett fält är `ReadWrite` eller ej. Om returvärdet är `TRUE` innebär det att det aktuella fältet är `ReadWrite`. Om returvärdet är `FALSE` innebär det att det aktuella fältet är `ReadOnly`. (`TRUE = 1`, `FALSE = 0` enligt definition i windows headerfiler). Funktionen `AdkIsReadWrite()` tar två inparametrar:

- `ADK_DATA* pData` – En pekare till den aktuella datastrukturen.
- `INT iFieldId` – Id till det fält där rättighet ska kontrolleras.

AdkSetWarningFunction (AdkSetWarningDelegate)

C++

```
VOID AdkSetWarningFunction(void(*function) (ADK_ERROR*))
```

C#

```
void AdkSetWarningDelegate(AdkWarningDelegate pFunction)
```

Information

Funktionen `AdkSetWarningFunction()` anger vilken funktion som ska anropas när en varning uppstår. En varning som uppstår i API:et hanteras inte internt i API:et, däremot skickas en datastruktur till den funktion som är avsedd för att ta hand om varningar, ifall en sådan har angetts med `AdkSetWarningFunction()`. Den datastruktur som skickas från API:et till den angivna funktionen innehåller information om den varning som har uppstått. En varning som skickas från API:et är endast en upplysning, det är ej möjligt att avbryta den aktuella operationen som har skapat varningen. Nedan följer ett exempel på hur en funktion för att ta emot varningar kan deklareras.

```
Void __stdcall LoggWarnings(ADK_ERROR* err)
```

Funktionen `LoggWarnings()` är nu möjlig att skicka till `AdkSetWarningFunction()` för att ange som den funktion där varningar kommer att hamna.

AdkGetVariant

C++

```
ADK_ERROR AdkGetVariant(INT* piVariant, BOOL* pbApi, CHAR** chBuf, INT  
iBufSize, CHAR* pszSystemPath)
```

C#

```
ADKERROR AdkGetVariant(out int piVariant, out bool pbApi, out string  
strBuf, string strSystemPath)
```

Information

Visma Integration för Visma Administration 500/1000/2000 och **Visma Integration för Visma Förening** använder alla filen `adk.dll`. Dessa program innehåller olika moduler och funktionalitet, tex finns offerthantering bara i **Visma Administration 2000**. Det kan därför vara nödvändigt att ta reda på vilket program som licensen avser så att ens egna program kan anpassas till det program som finns installerat hos slutanvändaren. Funktionen ersätter `AdkGetAdmSize()` resp `AdkGetAdmSizeEx()`, vilka inte bör användas från version 5.0.

Funktionen `AdkGetVariant()` hämtar information ur `adk.dll` om vilket program som är installerat. Detta kan erhållas antingen som ett heltalsvärde eller som en textsträng, beroende på medskickade inparametrar.

- `INT* piVariant` – En `int`-pekare som är definierad utanför API:et. Värdet kommer att skrivas där. `NULL` är tillåtet att ange, vilket gör att inget värde skrivs in.
- `BOOL* pbApi` – En `int`-pekare som är definierad utanför API:et. Värdet kommer att skrivas där. `NULL` är tillåtet att ange, vilket gör att inget värde skrivs in. Innehållet i fältet som pekas ut ska vara `TRUE` eller `FALSE`. (`TRUE` = 1, `FALSE` = 0 enligt definition i windows headerfiler).
- `CHAR** chBuf` – Minnesadressen till en `char`-pekare som är definierad utanför API:et. Namnet på den variant licensen avser kommer att skrivas in där. `NULL` är tillåtet att ange, vilket gör att inget värde skrivs in. Minst 44 tecken bör reserveras för resultatet utanför API:et för att hela texten ska få plats.
- `INT iBufSize` – storleken på textbufferten som reserverats för resultatet. Om `chBuf` är `NULL` saknar denna parameter betydelse.
- `CHAR* pszSystemPath` – en sträng som pekar ut var **Visma Administration 500/1000/2000** respektive **Visma Förening** är installerade.

Om exempelvis **Visma Integration för Visma Administration 2000** är installerat skrivs till `piVariant` värdet `ADKI_ADMIN2000` och texten "Visma Integration för Visma Administration 2000" skrivs till `chBuf`. Värdet `TRUE` (1) skrivs till `pbApi`.

Saknas licens för **Visma Integration** sätts `pbApi` till `FALSE` (0).

Om licens finns för **Visma Integration** sätts `pbApi` till `TRUE` (1).

Möjliga kombinationer är:

- `ADKI_ADMIN2000` – "Visma Integration för Visma Administration 2000" – `TRUE`
- `ADKI_ADMIN1000` – "Visma Integration för Visma Administration 1000" – `TRUE`
- `ADKI_ADMIN500` – "Visma Integration för Visma Administration 500" – `TRUE`
- `ADKI_FORENING` – "Visma Integration för Visma Förening" – `TRUE`
- `ADKI_ADMIN2000` – "Visma Administration 2000" – `FALSE`
- `ADKI_ADMIN1000` – "Visma Administration 1000" – `FALSE`
- `ADKI_ADMIN500` – "Visma Administration 500" – `FALSE`
- `ADKI_ADMIN200` – "Visma Administration 200" – `FALSE`

- ADKI_FORENING - "Visma Förening" - FALSE

AdkGetVariant() returnerar en datastruktur av typen ADK_ERROR, vilken innehåller information om resultatet av funktionen. Om felkod erhålls skrivs inga värden till piAdmSize respektive chBuf. Se avsnittet ADK_ERROR för ytterligare information.

AdkGetVariant() finns i AdkNet4Wrapper i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetCustomerPrice

C++

```
ADK_ERROR AdkGetCustomerPrice(CHAR* strCustomer, CHAR* strArticle, DOUBLE
dQuantity, BOOL bInvDisc, BOOL bRowDisc, BOOL bInclVAT, INT iSelCur,
DOUBLE* dVal)
```

C#

```
ADKERROR AdkGetCustomerPrice(string strCustomer, string strArticle,
double dQuantity, bool bInvRab, bool bRowRab, bool bInclVAT,
ADK_CODE_OF_CUSTOMERPRICE iSelCur, out double dVal)
```

Information

Funktionen är tänkt för beräkning av kundspecifika priser. Det finns ett antal parametrar i Visma Administration som påverkar en kunds pris. De parametrar som kan påverka priset är rabattavtal, prislistor, fakturarabatt samt radrabatt, där det under prislistor går att ange stafflingpriser och under rabattavtal går att ange rabatter på både artikelgruppsnivå samt artikelnivå. För att underlätta hämtningen av ett kundspecifikt pris har denna funktion lagts till. Funktionen tar åtta inparametrar

- CHAR* strCustomer – En charpekare som innehåller kundnumret för vilket pris ska beräknas.
- CHAR* strArticle – En charpekare som innehåller artikelnumret för vilket pris ska beräknas.
- DOUBLE dQuantity – Antalet artiklar som är tänkta att säljas (för att beräkna artikelns stafflingpris).
- BOOL bInvDisc – Om kundens eventuella fakturarabatt ska användas vid beräkningen eller inte. Innehållet i variablen ska vara TRUE eller FALSE. (TRUE = 1, FALSE = 0 enligt definition i windows headerfiler).
- BOOL bRowDisc – Om kundens eventuella radrabatter ska tas med vid beräkningen eller inte. Innehållet i variablen ska vara TRUE eller FALSE. (TRUE = 1, FALSE = 0 enligt definition i windows headerfiler).
- BOOL bInclVat – Om priset ska vara inklusive eller exklusive moms. Innehållet i variablen ska vara TRUE eller FALSE. (TRUE = 1, FALSE = 0 enligt definition i windows headerfiler).

- INT iSelCur – I vilket pris beräkningen ska ske
 - 0 – Kundens valuta
 - 1 – Inhemsk valuta
 - 2 – Slutgiltig prislistas valuta
- DOUBLE* dVal – Minnesadress där värdet kommer att skrivas.

AdkGetCustomerPrice() finns i *AdkNet4Wrapper* i två varianter. Den äldre har deklarerats som *deprecated* i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är *AdkNet4Wrapper.dll* bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetCustomerPrice2

C++

```
ADK_ERROR AdkGetCustomerPrice2(CHAR* strCustomer, CHAR* strArticle,
DOUBLE dQuantity, BOOL bInvDisc, BOOL bRowDisc, BOOL bInclVAT, INT
iSelCur, DOUBLE* dVal, CHAR** ppsCurrency, INT iLenCurrency)
```

C#

```
ADKERROR AdkGetCustomerPrice2(string strCustomer, string strArticle,
double dQuantity, bool bInvRab, bool bRowRab, bool bInclVAT,
ADK_CODE_OF_CUSTOMERPRICE iSelCur, out double dVal, out string
ppsCurrency)
```

Information

Funktionen är tänkt för beräkning av kundspecifika priser. Det finns ett antal parametrar i Visma Administration som påverkar en kunds pris. De parametrar som kan påverka priset är rabattavtal, prislistor, fakturarabatt samt radrabatt, där det under prislistor går att ange stafflingpriser och under rabattavtal går att ange rabatter på både artikelgruppsnivå samt artikelnivå. För att underlätta hämtningen av ett kundspecifikt pris har denna funktion lagts till. Utökningen mot *AdkGetCustomerPrice* är att valutakoden för den prislista som använts i beräkningen kan erhållas. Funktionen tar tio inparametrar

- CHAR* strCustomer – En charpekare som innehåller kundnumret för vilket pris ska beräknas.
- CHAR* strArticle – En charpekare som innehåller artikelnumret för vilket pris ska beräknas.
- DOUBLE dQuantity – Antalet artiklar som är tänkta att säljas (för att beräkna artikelns stafflingpris).
- BOOL bInvDisc – Om kundens eventuella fakturarabatt ska användas vid beräkningen eller inte. Innehållet i variablen ska vara TRUE eller FALSE. (TRUE = 1, FALSE = 0 enligt definition i windows headerfiler).
- BOOL bRowDisc – Om kundens eventuella radrabatter ska tas med vid beräkningen eller inte. Innehållet i variablen ska vara TRUE eller FALSE. (TRUE = 1, FALSE = 0 enligt definition i windows headerfiler).

- **BOOL blncIVat** – Om priset ska vara inklusive eller exklusive moms. Innehållet i variabeln ska vara **TRUE** eller **FALSE**. (**TRUE** = 1, **FALSE** = 0 enligt definition i windows headerfiler).
- **INT iSelCur** – I vilket pris beräkningen ska ske
 - 0 – Kundens valuta
 - 1 – Inhemsk valuta
 - 2 – Slutgiltig prislistas valuta
- **DOUBLE* dVal** – Minnesadress där värdet kommer att skrivas.
- **CHAR** ppsCurency** – Pekare till strängbuffert där valutakod skall sparas.
- **INT iLenCurrency** – Storlek på strängbuffert för valutakod.

AdkGetCustomerPrice2() finns i *AdkNet4Wrapper* i två varianter. Den äldre har deklarerats som *deprecated* i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är *AdkNet4Wrapper.dll* bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetVatpercentOnArticle

C++

```
ADK_ERROR AdkGetVatpercentOnArticle(CHAR* strArticle, DOUBLE* dVal)
```

C#

```
ADKERROR AdkGetVatpercentOnArticle(string strArticle, out double dVal)
```

Information

Funktionen är tänkt att användas för att hämta momssatsen för en specifik artikel i artikelregistret
Funktionen tar två inparametrar

- **CHAR* strArticle** – En charpekare som innehåller det artikelnummer för vilken momssats önskas.
- **DOUBLE* dVal** – Minnesadress där värdet kommer att skrivas.

AdkVatPercentOnArticle() finns i *AdkNet4Wrapper* i två varianter. Den äldre har deklarerats som *deprecated* i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är *AdkNet4Wrapper.dll* bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetSie

C++

```
ADK_ERROR AdkGetSie(ADK_SIE_TYPE type, CHAR chAar, CHAR* chFilename,  
CHAR* chPath)
```

C#

```
ADKERROR AdkGetSie(ADK_SIE_TYPE type, string strAar, string strFilename, string strPath)
```

Information

Funktionen är tänkt att användas för att skapa SIE-filer på motsvarande sätt som finns i Visma Administration och Visma Förening.

- ADK_SIE_TYPE type – Önskad SIE-variant.
- CHAR chAar – Bokföringsår som informationen ska baseras på.
- CHAR* chFilename – Filnamn på skapad fil.
- CHAR* chPath – Sökväg där filen ska skapas.

ADK_SIE_TYPE finns definierad i Adk.h.

AdkGetSie() finns i AdkNet4Wrapper i två varianter. Den äldre har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetBookingYearId

C++

```
ADK_ERROR AdkGetBookingYearId(LONG lValue, CHAR** chBuf, INT iBufSize)
```

C#

```
ADKERROR AdkGetBookingYearId(DateTime dValue, out string strId)
```

Information

Funktionen är tänkt att användas för att hämta id för bokföringsår som datumet tillhör.

- LONG lValue – Datum i formatet long (Julian).
- CHAR** chBuf – Pekare till strängbuffert där resultatet skall sparas.
- INT iBufSize – storlek på strängbufferten. Två tecken krävs för att svaret ska kunna lagras.

Id för bokföringsår kan vara A till Z och svaret levereras i en null-terminerad sträng. Funktionen kräver att databasen är öppen.

AdkGetBookingYearId() returnerar en datastruktur av typen ADK_ERROR, vilken innehåller information om resultatet av funktionen. En felkod, ADKE_DATE_NOT_WITHIN_ANY_BOOKING_YEAR,

är aktuell och om denna erhålls skrivs inga värden till `chBuf`. Se avsnittet `ADK_ERROR` för ytterligare information.

`AdkGetBookingYearId()` finns i `AdkNet4Wrapper` i två varianter. Den äldre har deklarerats som `deprecated` i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

`AdkGetVersion (AdkGetProgramVersion)`

C++

```
VOID AdkGetVersion(LONG* lMajor, LONG* lMinor)
```

C#

```
void AdkGetProgramVersion(out int lMajor, out int lMinor)
```

Information

Funktionen returnerar versionsinformation.

- `LONG* lMajor` – Pekare till longvariabel där major version ska lagras.
- `LONG* lMinor` – Pekare till longvariabel där minor version ska lagras.

Om resp parameter inte är `NULL` skrivs versionsinformation till resp parameter.

`AdkGetVersion()` har deklarerats som `deprecated` i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten `AdkGetProgramVersion()` ersätter funktionen i .NET. Dock är `AdkNet4Wrapper.dll` bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

`AdkGetCompanyName`

C++

```
ADK_ERROR AdkGetCompanyName(CHAR* pszFtgPath, BOOL bStrict, CHAR* pszSystemPath, CHAR** chBuf, INT iBufSize, INT* piVariant, LONG* lMajor, LONG* lMinor)
```

C#

```
ADKERROR AdkGetCompanyName(string pszFtgPath, bool bStrict, string strSystemPath, out string chBuf, out int piVariant, out int lMajor, out int lMinor)
```

Information

Funktionen returnerar företagsnamn samt databasversion för företaget. Kontroll att licens för integration finns genomförs.

- `CHAR* pszFtgPath` – sökväg till företaget

- **BOOL bStrict** – om satt till TRUE returneras felkod när företaget senast är öppnat i annan variant av programmet och inte kan öppnas av API:et. Innehållet i variablen ska vara TRUE eller FALSE. (TRUE = 1, FALSE = 0 enligt definition i windows headerfiler).
- **CHAR* pszSystemPath** – sökväg till gemensamma filer
- **CHAR** chBuf** – buffert där företags namn returneras
- **INT iBufSize** – storlek på buffert, minst 51 tecken rekommenderas
- **LONG* IMajor** – Pekare till longvariabel där major version ska lagras
- **LONG* IMinor** – Pekare till longvariabel där minor version ska lagras

NULL är tillåtet för chBuf, IMajor resp IMinor. I dessa fall skrivs inte information till resp parameter.

Programversion och databasversion from version 5.3:

Programversion	Databasversion
5.3 – 5.31	5.30
5.4 – 5.52	5.40
2013.0	13.00
2014.0 – 2014.02	14.00
2014.1 – 2014.12	14.10
2015.0	15.00
2015.1	15.10
2015.2 – 2016.1	15.20
2016.2 – 2016.23	16.20
2017.0	17.00
2017.1-2017.12	17.10
2018.0-2018.03	18.00
2018.1	18.10
2018.2	18.20
2018.3	18.30
2018.4	18.40
2018.5	18.50
2018.6	18.60
2018.7-2018.71	18.70
2019.0	19.00
2019.1	19.10
2019.2	19.20
2019.21	19.21
2019.3	19.30
2020.0	20.0

Funktionen kan returnera följande felkoder:

ADKE_INVALID_SERIAL
ADKE_INVALID_ADM_PATH
ADKE_INVALID_PATH_COMPANY

```
ADKE_INVALID_DB_PROG
ADKE_DB_NOT_ADM_2000
ADKE_DB_NOT_ADM_1000
ADKE_DB_NOT_ADM_500
ADKE_DB_NOT_ADM_FOR
```

Fr.o.m version 2013.0 kan med hjälp av funktionen `AdkGetProgramDatabaseVersion()` avgöras om företagets databas kan öppnas med aktuell version av APllet. Se nedan i avsnittet om `AdkGetProgramDatabaseVersion()`.

AdkGetCompanyName() finns i *AdkNet4Wrapper* i två varianter. Den äldre har deklarerats som *deprecated* i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är *AdkNet4Wrapper.dll* bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetCompanyNameEx

C++

```
ADK_ERROR AdkGetCompanyNameEx(CHAR* pszFtgPath, BOOL bStrict, CHAR*
pszSystemPath, CHAR** chBuf, INT iBufSize, INT* piVariant, LONG* lMajor,
LONG* lMinor)
```

C#

```
ADKERROR AdkGetCompanyNameEx(string pszFtgPath, bool bStrict, string
strSystemPath, out string chBuf, out int piVariant, out int lMajor, out
int lMinor)
```

Information

Funktionen returnerar företagsnamn samt databasversion för företaget. Kan användas även om licens för Integration saknas (fungera dock inte i Visma Administration 200).

- CHAR* pszFtgPath – sökväg till företaget
- BOOL bStrict – om satt till TRUE returneras felkod när företaget senast är öppnat i annan variant av programmet och inte kan öppnas av API:et. Innehållet i variablen ska vara TRUE eller FALSE. (TRUE = 1, FALSE = 0 enligt definition i windows headerfiler).
- CHAR* pszSystemPath – sökväg till gemensamma filer
- CHAR** chBuf – buffert där företagets namn returneras
- INT iBufSize – storlek på buffert, minst 51 tecken rekommenderas
- LONG* lMajor – Pekare till longvariabel där major version ska lagras
- LONG* lMinor – Pekare till longvariabel där minor version ska lagras

NULL är tillåtet för chBuf, lMajor resp lMinor. I dessa fall skrivs inte information till resp parameter.

Funktionen kan returnera följande felkoder:

```
ADKE_INVALID_SERIAL  
ADKE_INVALID_ADM_PATH  
ADKE_INVALID_PATH_COMPANY  
ADKE_INVALID_DB_PROG  
ADKE_DB_NOT_ADM_2000  
ADKE_DB_NOT_ADM_1000  
ADKE_DB_NOT_ADM_500  
ADKE_DB_NOT_ADM_FOR
```

AdkGetCompanyNameEx() finns i *AdkNet4Wrapper* i två varianter. Den äldre har deklarerats som *deprecated* i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är *AdkNet4Wrapper.dll* bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetCompanySetting

C++

```
ADK_ERROR AdkGetCompanySetting(CHAR* pszSetting, INT* piValue)
```

C#

```
ADKERROR AdkGetCompanySetting(string pszSetting, out bool bValue)
```

Information

Funktionen hämtar värdet för vissa företagsinställningar.

- CHAR* pszSetting – Sträng som representerar sökt inställning.
- INT* piValue – Pekare till intvariabel där värdet för inställningen ska sparas.

Funktionen returnerar ADKE_INVALID_STRING om pszSetting är felaktig och ADKE_INVALID_DATA_TYPE om piValue är satt till NULL.

Funktionen kräver att ett företag öppnats med AdkOpen, i annat fall returneras ADKE_NO_DB_OPEN.

Följande strängvärden kan anges:

- | | |
|----------------------|---|
| • "PRICE4DECIMALS" | – 4 decimaler i å-pris aktiverat |
| • "VERSERIES" | – Verifikationsserier aktiverat |
| • "EDI" | – Edi aktiverat |
| • "WEBSHOP" | – Webbshop aktiverat |
| • "BUILDSERVICE" | – Omvänd skattskyldighet aktiverat |
| • "TAXREDUCTION" | – Skattereduktion för husarbetet aktiverat |
| • "CURRENCYCLIENT" | – Valuta/språk aktiverat för kundreskontra |
| • "CURRENCYSUPPLIER" | – Valuta/språk aktiverat för leverantörsreskontra |

Värdet i piValue kommer att sättas till 0 om efterfrågad inställning är inaktiverad i företaget och 1 om den är aktiverad.

AdkGetCompanySetting() finns i *AdkNet4Wrapper* i två varianter. Den äldre har deklarerats som *deprecated* i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används.

Funktionen är inte relevant då den nya varianten ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

AdkGetProgramDatabaseVersion

C++

```
VOID AdkGetProgramDatabaseVersion(LONG* lMajor, LONG* lMinor)
```

C#

```
void AdkGetSupportedDatabaseVersion(out int lMajor, out int lMinor)
```

Information

Funktionen returnerar databasversionsinformation, dvs den version i företagens databas som stöds av API:et. Versionen är inte alltid samma som programmets version.

- LONG* lMajor – Pekare till longvariabel där major version ska lagras.
- LONG* lMinor – Pekare till longvariabel där minor version ska lagras.

Om resp parameter inte är NULL skrivs versionsinformation till resp parameter.

Se avsnittet ovan om AdkGetCompanyName() för information om databasversioner i olika programversioner.

AdkGetProgramDatabaseVersion() har deklarerats som deprecated i .NET. Detta innebär att vi rekommenderar att denna funktion inte längre används. Funktionen är inte relevant då den nya varianten AdkGetSupportedDatabaseVersion() ersätter funktionen i .NET. Dock är AdkNet4Wrapper.dll bakåtkompatibel och gamla funktionen kommer fortsätta att fungera som innan.

Funktioner som är ersatta med nyare versioner

AdkGetAdmSize

OBS! Denna funktion är obsolet. Använd AdkGetVariant istället.

C++

```
ADK_ERROR AdkGetAdmSize(INT* piAdmSize, CHAR** chBuf, INT iBufSize)
```

Information

Visma Integration för Visma Administration 500/1000/2000 och Visma Integration för Visma Förening använder alla filen adk.dll. Dessa program innehåller olika moduler och funktionalitet, tex finns offerthantering bara i **Visma Administration 2000**. Det kan därför vara nödvändigt att ta reda på vilket program licensen avser så att ens egna program kan anpassas till det program som finns installerat hos slutanvändaren.

Funktionen `AdkGetAdmSize()` hämtar information ur `adk.dll` om vilket program som är installerat. Detta kan erhållas antingen som ett heltalsvärde eller som en textsträng, beroende på medskickade inparametrar.

- `INT* piAdmSize` – En `int`-pekare som är definierad utanför API:et. Värdet kommer att skrivas där. `NULL` tillåtet.
- `CHAR** chBuf` – Minnesadressen till en `char`-pekare som är definierad utanför API:et. Värdet kommer att skrivas in där. `NULL` tillåtet. Minst 44 tecken bör reserveras för resultatet utanför API:et för att hela texten ska få plats.
- `INT iBufSize` – storleken på textbufferten som reserverats för resultatet. Om `chBuf` är `NULL` saknar denna parameter betydelse.

Om exempelvis **Visma Integration för Visma Administration 2000** är installerat skrivs till `piAdmSize` värdet `ADKI_ADMIN2000` och texten "Visma Integration för Visma Administration 2000" skrivs till `chBuf`.

Övriga värden är:

- `ADKI_ADMIN1000` – "Visma Integration för Visma Administration 1000"
- `ADKI_ADMIN500` – "Visma Integration för Visma Administration 500"
- `ADKI_FORENING` – "Visma Integration för Visma Förening"

`AdkGetAdmSize()` returnerar en datastruktur av typen `ADK_ERROR`, vilken innehåller information om resultatet av funktionen. Endast en felkod, `ADKE_INVALID_SERIAL`, är aktuell och om denna erhålls skrivs inga värden till `piAdmSize` respektive `chBuf`. Se avsnittet `ADK_ERROR` för ytterligare information.

Bilagor

Kodbeskrivning

Nedan följer en genomgång av de funktioner som API:et erbjuder. Till varje del finns ett kodexempel som beskrivs ingående. I utvecklarpaketet finns även ett demonstrationsexempel där API:et har använts mot programdelarna Kund och Faktura. Kodexemplen är skrivna i C++.

Denna bilaga utgår från att användaren av API:et definierat en klass för felhantering, `CerrorHandling`. Sist i bilagan finns ett litet exempel på hur det är möjligt att i klassen hantera felmeddelanden.

AdkOpen

För att öppna en koppling till en databas och **API:et** används funktionen `AdkOpen()`, denna funktion tar som inparametrar sökvägen till programmet samt sökvägen till det företag eller den förening som ska öppnas.

Först skapas en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter skapas två textsträngar med sökvägar till programmets gemensamma filer för **Visma Administration 500/1000/2000** respektive **Visma Förening** och det företag eller förening som arbete ska ske med.

```
Char* pcSys = new char[200];
char* pcFtg = new char[200];
strcpy(pcSys, "C:\\ProgramData\\SPCS\\SPCS Administration\\Gemensamma filer");
strcpy(pcFtg, "C:\\ProgramData\\SPCS\\SPCS Administration\\Företag\\Ovnbol2000");

error = AdkOpen(pcSys, pcFtg);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

När strängarna är angivna anropas `AdkOpen()` som returnerar en datastruktur av typen `ADK_ERROR`, här anges om något har gått fel. Serienummer, licensnummer och antalet licenser kommer att valideras när `AdkOpen()` anropas.

AdkSetWarningFunction

Varningar anges inte som fel från API:et. Varningar kommer att skickas till en funktion utanför API:et. För att ange en funktion där varningar ska tas emot används funktionen `AdkSetWarningFunction()`. Om ingen funktion för felhantering kopplats på detta sätt, kommer varningar inte att rapporteras utanför API:et. Den funktion som ska kopplas till varningsmeddelanden ska vara av typen `void` och ta en pekare till en `ADK_ERROR`-struktur som inparameter.

```
Void _stdcall Warning(ADK_ERROR* error)
AdkSetWarningFunction(Warning);
```

Funktionen som skapas för att ta hand av varningar måste deklarerars enligt ovan. Observera att här, i likhet med funktionsdeklarationerna i `adk.h`, deklarerars funktionen med `_stdcall` som calling convention.

AdkAdd

Lägga till Artikel

För att skapa en artikel används funktionen `AdkAdd()`. Det första som ska göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;  
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur av önskad typ skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_ARTICLE` för att tala om att det är en datastruktur av typen "Artikel" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_ARTICLE);
```

Med datastrukturen `pData` skapad, kan fält i strukturen tilldelas värden. Detta görs med de olika `AdkSet`-funktionerna. I bilagan **Registerbeskrivning** finns alla fält som är möjliga att arbeta med, t ex motsvaras artikelnummer av fältet `ADK_ARTICLE_NUMBER`. I bilagan **Registerbeskrivning** framgår även att fältet är av typen `character`, d v s en sträng. Denna information kan även fås med hjälp av funktionen `AdkGetType()`. För att kontrollera fältets längd kan funktionen `AdkGetLength()` användas. Funktionen `AdkSetStr()` används för att tilldela ett värde till detta fält.

```
Error = AdkSetStr(pData, ADK_ARTICLE_NUMBER, "Artikell");  
if(error.lRc != ADKE_OK)  
{  
    CerrorHandling rErrorHandling(&error);  
}
```

Till funktionen `AdkSetStr()` skickas den datastruktur `pData` som tidigare skapats, det fält där ett värde ska sättas och det värde som ska sättas. Funktionen `AdkSetStr()` returnerar en datastruktur av typen `ADK_ERROR`. Denna struktur tas emot i den tidigare skapade datastrukturen `error`. Därefter valideras funktionen `error` för att se om något misslyckats. Valideringen sker genom att returkoden `lRc` jämförs med värdet för OK som är `ADKE_OK`. Den validering som beskrivs ovan kommer att ske efter varje funktion som anropas och returnerar en datastruktur av typen `ADK_ERROR`. När alla fält har tilldelats till datastrukturen anropas funktionen `AdkAdd()` för att skriva till databasen.

```
Error = AdkAdd(pData);  
if(error.lRc != ADKE_OK)  
{  
    CerrorHandling rErrorHandling(&error);  
}
```

Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som används.

```
AdkDeleteStruct(pData);
```

Lägga till Kund

För att skapa en kund används funktionen `AdkAdd()`. Det första som ska göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;  
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur av önskad typ skapas, vilket görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_CUSTOMER` för att tala om att det är en datastruktur av typen "Kund" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_CUSTOMER);
```

Med strukturen för en kund skapad, kan fält i datastrukturen tilldelas värden. Detta görs med de olika `AdkSet`-funktionerna. I bilagan **Registerbeskrivning** finns alla fält som är möjliga att arbeta med beskrivna, t ex motsvaras kundnummer av fältet `ADK_CUSTOMER_NUMBER`. I registerbeskrivning framgår även att fältet är av typen `character`, alltså en sträng. Denna information kan även fås med hjälp av funktionen `AdkGetType()`. Funktionen `AdkSetStr()` används för att tilldela ett värde till detta fält.

```
Error = AdkSetStr(pData, ADK_CUSTOMER_NUMBER, "Kund1");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Till funktionen `AdkSetStr()` skickas den datastruktur `pData` som tidigare skapats, det fält där ett värde ska sättas och det värde som önskas sättas. Funktionen `AdkSetStr()` returnerar en datastruktur av typen `ADK_ERROR`. Denna struktur tas emot i den tidigare skapade datastrukturen `error`. Därefter valideras funktionen `error` för att se om något misslyckats. Valideringen sker genom att returkoden `lRc` jämförs med värdet för OK som är `ADKE_OK`. Den validering som beskrivs ovan kommer att ske efter varje funktion som anropas och returnerar en datastruktur av typen `ADK_ERROR`. När alla fält har tilldelats till datastrukturen anropas funktionen `AdkAdd()` för att skriva till databasen.

```
Error = AdkAdd(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som används.

```
AdkDeleteStruct(pData);
```

Lägga till faktura

För att skapa en faktura används funktionen `AdkAdd()`. Det första som ska göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur av önskad typ skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_INVOICE_HEAD` för att tala om att det är en datastruktur av typen "Fakturahuvud" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_INVOICE_HEAD);
```

Med strukturen för ett fakturahuvud skapat, kan fält i datastrukturen tilldelas värden. Detta sker med de olika `AdkSet`-funktionerna. I bilagan **Registerbeskrivning** beskrivs alla fält och dess egenskaper. API:ets hjälpfunktioner kan också användas, för att hämta egenskaper för fält.

Först anges för vilket kundnummer som fakturan ska skapas. Kundnumret på en faktura motsvaras av fältet `ADK_OOI_HEAD_CUSTOMER_NUMBER`. För att ta reda på fältets typ anropas funktionen `AdkGetType()`. För att använda denna funktion krävs att en variabel av typen `ADK_FIELD_TYPE` har deklarerats.

```
ADK_FIELD_TYPE iType;
error = AdkGetType(pData, ADK_OOI_HEAD_CUSTOMER_NUMBER, &iType);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Variabeln `iType` kommer att anta värdet för fältets typ, i detta fall är typen `eChar`. För att tilldela ett fält av typen `eChar` ett värde används funktionen `AdkSetStr()`. Innan skrivning sker till fältet bör en

kontroll göras om det är tillåtet att skriva till fältet eller om fältet endast är till för läsning. Funktionen `AdkIsReadWrite()` används för denna kontroll. Om svaret från funktionen `AdkIsReadWrite()` är sant kommer en skrivning till datastrukturen med `AdkSetStr()` att ske i vårt exempel.

```
If(AdkIsReadWrite(pData, ADK_OOI_HEAD_CUSTOMER_NUMBER))
{
    error = AdkSetStr(pData, ADK_OOI_HEAD_CUSTOMER_NUMBER, "Kund1");
    if(error.lRc != ADKE_OK)
    {
        CerrorHandling rErrorHandling(&error);
    }
}
```

Till funktionen `AdkSetStr()` skickas den datastruktur `pData` som tidigare skapats, det fält där ett värde önskas sättas och det värde som önskas sättas.

Att hämta ett fälts typ och att kontrollera om det är skrivbart eller ej bör ske före varje skrivning till datastrukturen. Exemplet som följer utgår från att detta har skett.

När ett kundnummer har satts på fakturan kan nästa steg vara att t ex sätta ett värde som anger att fakturan är en exportfaktura. För att ange detta sätts fältet `ADK_OOI_HEAD_EXPORT` till sant med hjälp av funktionen `AdkSetBool()`.

```
Error = AdkSetBool(pData, ADK_OOI_HEAD_EXPORT, 1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Till funktionen `AdkSetBool()` skickas den datastruktur `pData` som tidigare skapats, det fält där ett värde önskas sättas och det värde som önskas sättas. `AdkSetBool()` tar en `INT` som inparameter nummer tre. En `INT` behandlas i databasen som en `int`, där ett betyder sant och noll betyder falskt.

Efter att exportflaggan har fått ett värde sätter vi ett fakturadatum. Fältet för fakturadatum är `ADK_OOI_HEAD_DOCUMENT_DATE1` och är av typen `eDate`. Ett datum i databasen behandlas som ett `long`-värde (Juliandatum). Därför måste ett datum i en textsträng konverteras till ett datum i form av en `long`. Denna konvertering utförs med funktionen `AdkDateToLong()`. I exemplet nedan ska datumet "2001-11-01" vara fakturadatum. För att ta emot datumet i form av en `long` behövs en variabel, `lDate`.

```
Long lDate;
error = AdkDateToLong("2001-11-01", &lDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Variabeln `lDate` innehåller nu det `long`-värde som motsvarar angivet datum och det är detta värde som ska skickas till funktionen `AdkSetDate()`, vilken tar en `long` som inparameter nummer tre. Vid felaktigt datum erhålls dagens datum.

```
Error = AdkSetDate(pData, ADK_OOI_HEAD_DOCUMENT_DATE1, lDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Alla värden som anges till en post kommer att skrivas till databasen i nummerordning. Fältet med index noll kommer att skrivas först och så vidare. Detta medför att även kontroller kommer att utföras i denna ordning. Om t ex fakturan sätts till utskriven kommer det ej bli möjligt att sätta fler värden på fakturan. Om värden tilldelas efter att flaggan för utskriven har satts till sann kommer ett felmeddelande att returneras. Om fakturan ska fyllas med värden och samtidigt sättas till utskriven måste detta ske i två steg. Först skapas fakturan med alla värden och därefter sätts, med hjälp av en uppdatering, flaggan för utskriven till sann på fakturan.

När fakturadatomet är satt är fakturahuvudet klart och då återstår att skapa rader till huvudet. Skapandet av rader sker med funktionen `AdkCreateDataRow()`, som skapar typ och antal av rader som önskas. I exemplet nedan ska två rader av typen "Fakturarader" skapas.

```
Int nRows = 2;
PADK_DATA pRowData = AdkCreateDataRow(ADK_DB_INVOICE_ROW, nRows);
```

Två fakturarader har skapats och pekaren `pRowData` pekar på dessa, det är alltså en vektor med två element som skapats. För att arbeta med en av dessa två rader skapas en temporär datastruktur som kommer att tilldelas en av raderna med funktionen `AdkGetDataRow()`.

```
PADK_DATA pTempData = AdkGetDataRow(pRowData, 0);
```

Den temporära datastrukturen `pTempData` pekare nu på raden med index 0, alltså den första raden. Raderna som skapats har nummer ett och två men index 0 och 1. Det är nu möjligt att tilldela rad nummer ett värden på samma sätt som huvudet tilldelades värden. Först tilldelas ett artikelnummer till raden. Artikelnumret på raden är fältet `ADK_OOI_ROW_ARTICLE_NUMBER` och är av typen `eChar`.

```
Error = AdkSetStr(pTempData, ADK_OOI_ROW_ARTICLE_NUMBER, "Artikell");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

När artikelnumret satts kan antalet beställda artiklar sättas. Antalet beställda artiklar motsvaras av fältet `ADK_OOI_ROW_QUANTITY2` och är av typen `eDouble`. Eftersom fältet är av typen `double` är det möjligt att det kan innehålla decimaler. Det antal decimaler fältet kan innehålla styrs av vilken enhet som är angiven för raden.

I exemplet nedan kommer antalet beställda artiklar att sättas till tio. För att tilldela värde till fältet används funktionen `AdkSetDouble()`.

```
Error = AdkSetDouble(pTempData, ADK_OOI_ROW_QUANTITY2, 10);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu är rad ett klar och nedan kommer rad två att tilldelas samma artikel men med antalet 20 som beställt antal.

```
pTempData = AdkGetDataRow(pRowData, 1);

error = AdkSetStr(pTempData, ADK_OOI_ROW_ARTICLE_NUMBER, "Artikell");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetDouble(pTempData, ADK_OOI_ROW_QUANTITY2, 20);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

När de värden som ska anges till fakturahuvudet och fakturaraderna har satts, ska huvudet kopplas samman med raderna. Först ska antalet rader anges till huvudet. Fältet för denna information är `ADK_OOI_HEAD_NROWS` och är av typen `eDouble`.

```
Error = AdkSetDouble(pData, ADK_OOI_HEAD_NROWS, nRows);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

För att koppla raderna till huvudet används funktionen `AdkSetData()`. Det är fältet `ADK_OOI_HEAD_ROWS` i huvudet som ska kopplas till raderna. Kopplingen består av den pekare som tidigare skapats till raderna `pRowData`. Fältet `ADK_OOI_HEAD_ROWS` i huvudet kommer att innehålla denna pekare.

```
Error = AdkSetData(pData, ADK_OOI_HEAD_ROWS, pRowData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu är initieringen av datastrukturen klar och funktionen `AdkAdd()` kan anropas för att skriva till databasen.

```
Error = AdkAdd(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu har fakturan skrivits till databasen.

Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som använts. När `AdkDeleteStruct()` anropas med fakturahuvudet `pData` som inparameter kommer även det minne som allokerats till raderna att avallokeras.

```
AdkDeleteStruct(pData);
```

Lägga till inkommande följesedel

För att skapa en inkommande följesedel används funktionen `AdkAdd()`. Det första som ska göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur av önskad typ skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_DELIVERY_NOTE_HEAD` för att tala om att det är en datastruktur av typen "Följesedelshuvud" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_DELIVERY_NOTE_HEAD);
```

Med strukturen för ett följesedelshuvud skapat, kan fält i datastrukturen tilldelas värden. Detta sker med de olika `AdkSet`-funktionerna. I bilagan **Registerbeskrivning** beskrivs alla fält och dess egenskaper. API:ets hjälpfunktioner kan också användas, för att hämta egenskaper för fält.

Först anges för vilket leverantörsnummer som följesedeln ska skapas. Leverantörsnumret på en följesedel motsvaras av fältet `ADK_DELIVERY_NOTE_SUPPLIER_NUMBER`. För att ta reda på fältets typ anropas funktionen `AdkGetType()`. För att använda denna funktion krävs att en variabel av typen `ADK_FIELD_TYPE` har deklarerats.

```
ADK_FIELD_TYPE iType;
error = AdkGetType(pData, ADK_DELIVERY_NOTE_SUPPLIER_NUMBER, &iType);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Variabeln `iType` kommer att anta värdet för fältets typ, i detta fall är typen `eChar`. För att tilldela ett fält av typen `eChar` ett värde används funktionen `AdkSetStr()`. Innan skrivning sker till fältet bör en kontroll göras om det är tillåtet att skriva till fältet eller om fältet endast är till för läsning. Funktionen `AdkIsReadWrite()` används för denna kontroll. Om svaret från funktionen `AdkIsReadWrite()` är sant kommer en skrivning till datastrukturen med `AdkSetStr()` att ske i vårt exempel.

```

If(AdkIsReadWrite(pData, ADK_DELIVERY_NOTE_SUPPLIER_NUMBER))
{
    error = AdkSetStr(pData, ADK_DELIVERY_NOTE_SUPPLIER_NUMBER, "Leverantör1");
    if(error.lRc != ADKE_OK)
    {
        CerrorHandling rErrorHandling(&error);
    }
}

```

Till funktionen `AdkSetStr()` skickas den datastruktur `pData` som tidigare skapats, det fält där ett värde önskas sättas och det värde som önskas sättas. Att hämta ett fälts typ och att kontrollera om det är skrivbart eller ej bör ske före varje skrivning till datastrukturen. Exempelen som följer utgår från att detta har skett.

Skapandet av rader sker med funktionen `AdkCreateDataRow()`, som skapar typ och antal av rader som önskas. Det går att arbeta med rader på samma sätt som exempelvis fakturor. I exemplet nedan är det beställningsrader som ska läsas upp på följesedeln.

```

Int nRows = 2;
PADK_DATA pRowData = AdkCreateDataRow(ADK_DB_DELIVERY_NOTE_ROW, nRows);

```

Två följesedelsrader har skapats och pekaren `pRowData` pekar på dessa, det är alltså en vektor med två element som skapats. För att arbeta med en av dessa två rader skapas en temporär datastruktur som kommer att tilldelas en av raderna med funktionen `AdkGetDataRow()`.

```

PADK_DATA pTempData = AdkGetDataRow(pRowData, 0);

```

Den temporära datastrukturen `pTempData` pekare nu på raden med index 0, alltså den första raden. Raderna som skapats har nummer ett och två men index 0 och 1. Det är nu möjligt att tilldela rad nummer ett de värden som är obligatoriska för att kunna läsa upp beställningsrader på följesedeln. Detta sker på samma sätt som huvudet tilldelades värden.

Först tilldelas en dokumenttyp till raden. Dokumenttyp på raden är fältet `ADK_OOI_ROW_FROM_TYPE` och är av typen `eChar`. Eftersom det är en beställning förväntas värdet 'B'.

```

error = AdkSetStr(pTempData, ADK_OOI_ROW_FROM_TYPE, "B");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Övriga fält som är obligatoriska är vilket dokumentnummer som beställningen har. Det motsvaras av fältet `ADK_OOI_ROW_FROM_DOCUMENT` och är av typen `eDouble`. I exemplet nedan är det rader från beställningen med löpnummer 10 som läses upp på följesedeln.

```

Error = AdkSetDouble(pTempData, ADK_OOI_ROW_FROM_DOCUMENT, 10);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

I detta är följesedeln förberedd för att rader från aktuell beställning ska kunna läsas in, men inga rader är ännu inlästa om raden skulle sparas som den är. Inläsning av själva raden sker genom tilldelning av fältet `ADK_OOI_ROW_FROM_DOCROW` som är av typen `eDouble`. I exemplet nedan läses beställningsraden med position 1 upp på följesedeln.

```

Error = AdkSetDouble(pTempData, ADK_OOI_ROW_FROM_DOCROW, 1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Positionsnumret för beställningsraderna exponeras via fältet `ADK_OOI_ROW_CONNECTION_DOCROW` som är av typen `eDouble`.

När beställningsraden är uppläst kan antalet levererade artiklar sättas. Antalet beställda artiklar motsvaras av fältet `ADK_OOI_ROW_QUANTITY1` och är av typen `eDouble`. Eftersom fältet är av typen `double` är det möjligt att det kan innehålla decimaler. Det antal decimaler fältet kan innehålla styrs av vilken enhet som är angiven för raden.

I exemplet nedan kommer antalet levererade artiklar att sättas till tio. För att tilldela värde till fältet används funktionen `AdkSetDouble()`.

```
Error = AdkSetDouble(pTempData, ADK_OOI_ROW_QUANTITY1, 10);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu är rad ett klar och nedan kommer rad två att läsas upp från samma beställning men med antalet 20 som levererat antal. Detta förutsätter att beställningen har två rader eller fler.

```
pTempData = AdkGetDataRow(pRowData, 1);

error = AdkSetStr(pTempData, ADK_OOI_ROW_FROM_TYPE, "B");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetDouble(pTempData, ADK_OOI_ROW_FROM_DOCUMENT, 10);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetDouble(pTempData, ADK_OOI_ROW_FROM_DOCROW, 2);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetDouble(pTempData, ADK_OOI_ROW_QUANTITY1, 20);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

När de värden som ska anges till följesedelshuvudet och följesedelsraderna har satts, ska huvudet kopplas samman med raderna. Först ska antalet rader anges till huvudet. Fältet för denna information är `ADK_DELIVERY_NOTE_NROWS` och är av typen `eDouble`.

```
Error = AdkSetDouble(pData, ADK_DELIVERY_NOTE_NROWS, nRows);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

För att koppla raderna till huvudet används funktionen `AdkSetData()`. Det är fältet `ADK_DELIVERY_NOTE_ROWS` i huvudet som ska kopplas till raderna. Kopplingen består av den pekare som tidigare skapats till raderna `pRowData`. Fältet `ADK_DELIVERY_NOTE_ROWS` i huvudet kommer att innehålla denna pekare.

```
Error = AdkSetData(pData, ADK_DELIVERY_NOTE_ROWS, pRowData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu är initieringen av datastrukturen klar och funktionen `AdkAdd()` kan anropas för att skriva till databasen.

```

Error = AdkAdd(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Nu har följesedeln skrivits till databasen.

Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som använts. När `AdkDeleteStruct()` anropas med följesedelshuvudet `pData` som inparameter kommer även det minne som allokerats till raderna att avallokeras.

```

AdkDeleteStruct(pData);

```

Lägga till kolli

För att skapa ett kolli till en order används funktionen `AdkAdd()`. Det första som ska göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```

ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));

```

Därefter ska en datastruktur av önskad typ skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_PACKAGE_HEAD` för att tala om att det är en datastruktur av typen "Kolli" som ska skapas.

```

PADK_DATA pData = AdkCreateData(ADK_DB_PACKAGE_HEAD);

```

Med strukturen för ett kolli skapat, kan fält i datastrukturen tilldelas värden. Detta sker med de olika `AdkSet`-funktionerna. I bilagan **Registerbeskrivning** beskrivs alla fält och dess egenskaper. API:ets hjälpfunktioner kan också användas, för att hämta egenskaper för fält.

För att kunna skapa ett kolli på en order krävs det att orderns löpnummer anges. Orderns löpnummer motsvaras av fältet `ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT`. För att ta reda på fältets typ anropas funktionen `AdkGetType()`. För att använda denna funktion krävs att en variabel av typen `ADK_FIELD_TYPE` har deklarerats.

```

ADK_FIELD_TYPE iType;
error = AdkGetType(pData, ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT, &iType);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Variabeln `iType` kommer att anta värdet för fältets typ, i detta fall är typen `eDouble`. För att tilldela ett fält av typen `eDouble` ett värde används funktionen `AdkSetDouble()`. Innan skrivning sker till fältet bör en kontroll göras om det är tillåtet att skriva till fältet eller om fältet endast är till för läsning. Funktionen `AdkIsReadWrite()` används för denna kontroll. Om svaret från funktionen `AdkIsReadWrite()` är sant kommer en skrivning till datastrukturen med `AdkSetStr()` att ske i vårt exempel.

```

If(AdkIsReadWrite(pData, ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT))
{
    error = AdkSetDouble(pData, ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT , 1);
    if(error.lRc != ADKE_OK)
    {
        CerrorHandling rErrorHandling(&error);
    }
}

```

Till funktionen `AdkSetDouble()` skickas den datastruktur `pData` som tidigare skapats, det fält där ett värde önskas sättas och det värde som önskas sättas. Att hämta ett fälts typ och att kontrollera om det är skrivbart eller ej bör ske före varje skrivning till datastrukturen. Exempelen som följer utgår från att detta har skett.

Skapandet av rader till kollit sker med funktionen `AdkCreateDataRow()`, som skapar typ och antal av rader som önskas. I exemplet nedan visas hur kollirader skapas.

```
Int nRows = 2;
PADK_DATA pRowData = AdkCreateDataRow(ADK_DB_PACKAGE_ROW, nRows);
```

Två kollirader har skapats och pekaren `pRowData` pekar på dessa, det är alltså en vektor med två element som skapats. För att arbeta med en av dessa två rader skapas en temporär datastruktur som kommer att tilldelas en av raderna med funktionen `AdkGetDataRow()`.

```
PADK_DATA pTempData = AdkGetDataRow(pRowData, 0);
```

Den temporära datastrukturen `pTempData` pekare nu på raden med index 0, alltså den första raden. Raderna som skapats har nummer ett och två men index 0 och 1. Det är nu möjligt att tilldela rad nummer ett de värden som är obligatoriska för att kunna skapa kolliraden. För att skapa en rad till ett kolli krävs det att orderraden anges. Detta sker via fältet `ADK_PACKAGE_ROW_ORDER_ROW_NUMBER`. Detta sker på samma sätt som huvudet tilldelades värden.

```
Error = AdkSetDouble(pTempData, ADK_PACKAGE_ROW_ORDER_ROW_NUMBER, 1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Positionsnumret för orderraderna exponeras via fältet `ADK_OOI_ROW_CONNECTION_DOCROW` som är av typen `eDouble`.

När kolliraden är skapad kan antalet artiklar sättas. Antalet beställda artiklar motsvaras av fältet `ADK_PACKAGE_ROW_QUANTITY` och är av typen `eDouble`. Eftersom fältet är av typen `double` är det möjligt att det kan innehålla decimaler. Det antal decimaler fältet kan innehålla styrs av vilken enhet som är angiven för raden.

I exemplet nedan kommer antalet levererade artiklar att sättas till tio. För att tilldela värde till fältet används funktionen `AdkSetDouble()`.

```
Error = AdkSetDouble(pTempData, ADK_PACKAGE_ROW_QUANTITY, 10);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu är rad ett klar och nedan kommer rad två att skapas med antalet 20 som levererat antal. Detta förutsätter att ordern har två rader eller fler.

```
pTempData = AdkGetDataRow(pRowData, 1);

error = AdkSetDouble(pTempData, ADK_PACKAGE_ROW_ORDER_ROW_NUMBER, 2);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetDouble(pTempData, ADK_PACKAGE_ROW_QUANTITY, 20);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

När de värden som ska anges till följesedelshuvudet och följesedelsraderna har satts, ska huvudet kopplas samman med raderna. Först ska antalet rader anges till huvudet. Fältet för denna information är `ADK_PACKAGE_HEAD_NROWS` och är av typen `eDouble`.

```
Error = AdkSetDouble(pData, ADK_PACKAGE_HEAD_NROWS, nRows);
if(error.lRc != ADKE_OK)
{
```

```

        CerrorHandling rErrorHandling(&error);
    }

```

För att koppla raderna till huvudet används funktionen `AdkSetData()`. Det är fältet `ADK_PACKAGE_HEAD_ROWS` i huvudet som ska kopplas till raderna. Kopplingen består av den pekare som tidigare skapats till raderna `pRowData`. Fältet `ADK_PACKAGE_HEAD_ROWS` i huvudet kommer att innehålla denna pekare.

```

Error = AdkSetData(pData, ADK_PACKAGE_HEAD_ROWS, pRowData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Nu är initieringen av datastrukturen klar och funktionen `AdkAdd()` kan anropas för att skriva till databasen.

```

Error = AdkAdd(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Nu har kollit skrivits till databasen.

Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som använts. När `AdkDeleteStruct()` anropas med kollihuvudet `pData` som inparameter kommer även det minne som allokerats till raderna att avallokeras.

```

AdkDeleteStruct(pData);

```

Lägga till underkolli

Att skapa underkolli skiljer sig inte mycket från att skapa kolli. För att skapa ett underkolli måste fältet `ADK_PACKAGE_HEAD_PACKAGE_REFERENCE` anges på kollit. I exemplet nedan skapas underkollit "1:1" till kollit "1".

```

Error = AdkSetStr(pData, ADK_PACKAGE_HEAD_PACKAGE_REFERENCE, "1");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

AdkUpdate

Uppdatera en kund

För att uppdatera en kund används funktionen `AdkUpdate()`. Det första som ska göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```

ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));

```

Därefter kan en datastruktur av önskad typ skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_CUSTOMER` för att tala om att det är en datastruktur av typen "Kund" som ska skapas.

```

PADK_DATA pData = AdkCreateData(ADK_DB_CUSTOMER);

```

Därefter ska den kund som ska uppdateras pekars ut. Detta sker genom att ett värde anges till nyckelfältet för databastabellen kund, vilket är kundnummer. Värdet som angivits kommer att fungera

som sökvärde, om ingen post motsvarar angivet värde kommer ett felmeddelande att returneras. Det är ej möjligt att uppdatera nyckelfältet.

```
Error = AdkSetStr(pData, ADK_CUSTOMER_NUMBER, "Kund1");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Till `AdkSetStr()` skickas datastrukturen `pData`, fältets id och det värde som ska anges. Ovan pekas ut att posten med kundnummer "Kund1" ska uppdateras. Efter att sökvärde satts ska de värden som ska uppdateras sättas. Kunden ska uppdateras med en ny procentsats för fakturarabatt. Fältet som motsvarar detta är `ADK_CUSTOMER_INVOICE_DISCOUNT`. Detta fält är av typen `eDouble`. Med `AdkSetDouble()` sätts det nya värdet i datastrukturen.

```
Error = AdkSetDouble(pData, ADK_CUSTOMER_INVOICE_DISCOUNT, 15);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Kunden kommer att få en ny fakturarabatt som är 15%. När alla fält som ska uppdateras har fått värden anropas funktion `AdkUpdate()` med datastrukturen `pData` för att uppdatera posten i databasen.

```
Error = AdkUpdate(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu har uppdateringen i databasen skett och kunden har fått en fakturarabatt på 15%. Därefter återstår bara att frigöra minne vilket görs med funktionen `AdkDeleteStruct()`.

```
AdkDeleteStruct(pData);
```

Uppdatera en faktura

För att uppdatera en faktura används funktionen `AdkUpdate()`. Det första som ska göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur av önskad typ skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_INVOICE_HEAD` för att tala om att det är en datastruktur av typen "Fakturahuvud" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_INVOICE_HEAD);
```

Med fakturahuvudet skapat är det dags att peka ut vilken faktura som ska uppdateras. Detta görs genom att tilldela ett värde till fakturans nyckelfält som är `ADK_OOI_HEAD_DOCUMENT_NUMBER`.

```
Error = AdkSetDouble(pData, ADK_OOI_HEAD_DOCUMENT_NUMBER, 1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

I raden ovan anges att det är faktura nummer ett (1) som ska uppdateras. Detta värde kommer att fungera som sökvärde bland fakturor. Finns ingen faktura med dokumentnummer ett (1) kommer ett felmeddelande att returneras. Först kommer fakturan att sökas upp, vissa av de värden fakturan innehåller sedan tidigare behövs för att göra uppdateringen.

```
Error = AdkSetSortOrder(pData, eInvoiceNr);
```

```

if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkFind(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Därefter ska de värden som ska uppdateras sättas. Om t ex fakturadatomet ska uppdateras kan det göras enligt följande:

```

long lDate;
error = AdkDateToLong("2002-10-15", &lDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkSetDate(pData, ADK_OOI_HEAD_DOCUMENT_DATE1, lDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

För att kunna uppdatera fakturadatomet anropas funktion `AdkDateToLong()` som konverterar ett datum i form av en sträng till ett `long`-värde (Juliandatum), vilket kan läggas till i databasen. När konverteringen har skett anropas `AdkSetDate()` som tilldelar ett värde av typen `long` till datastrukturen `pData`, som är datastrukturen för fakturahuvudet.

Det är även möjligt att uppdatera och skapa nya rader på en faktura. För att kunna uppdatera en rad eller lägga till en rad måste information om antalet rader finnas. Denna information fås genom att antalet rader läsas från fältet `ADK_OOI_HEAD_NROWS`. Detta är möjligt tack vare att fakturan tidigare söktes upp med `AdkFind()`.

```

Double nRows;
error = AdkGetDouble(pData, ADK_OOI_HEAD_NROWS, &nRows);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Nu innehåller variabeln `nRows` antalet rader. Om dessa två rader ska uppdateras behöver inga nya rader skapas. Raderna kom med datastrukturen vid `AdkFind()` ovan, och kan hämtas fram med hjälp av `AdkGetData()`. Om däremot en rad ska läggas till ska datastruktur för fler än två rader skapas. För att undvika minnesläckage måste först datastrukturen tömmas:

```

error = AdkResetStruct(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Eftersom vi har rensat datastrukturen behöver fakturanummer och datum sättas på samma sätt som tidigare i exemplet:

```

error = AdkSetDouble(pData, ADK_OOI_HEAD_DOCUMENT_NUMBER, 1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

long lDate;
error = AdkDateToLong("2002-10-15", &lDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkSetDate(pData, ADK_OOI_HEAD_DOCUMENT_DATE1, lDate);

```

```

if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Det antal över två som skapas kommer att bli nya rader på fakturan. I exemplet nedan läggs en rad till och en datastruktur med tre rader skapas således.

```

Int nNewRows = 3;
PADK_DATA pRowData = AdkCreateDataRow(ADK_DB_INVOICE_ROW, nNewRows);

```

Först uppdateras rad ett med ett nytt antal för beställda artiklar. Rad ett pekas ut med hjälp av funktionen `AdkGetDataRow()`, till funktionen skickas noll som index. Index noll motsvarar rad ett, index ett motsvarar rad två o s v. En temporär datastruktur `pTempRow` skapas för att hålla aktuell rad.

```

PADK_DATA pTempRow = AdkGetDataRow(pRowData, 0);
error = AdkSetDouble(pTempRow, ADK_OOI_ROW_QUANTITY2, 100);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Arbetet med den temporära datastrukturen `pTempRow` sker på samma sätt som arbetet med t ex ett dokumenthuvud. I exemplet nedan ska även antalet beställda av varan uppdateras på rad nummer två och därför pekas index ett ut och det blir möjligt att arbeta med rad nummer två.

```

pTempRow = AdkGetDataRow(pRowData, 1);
error = AdkSetDouble(pTempRow, ADK_OOI_ROW_QUANTITY2, 200);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Därefter ska arbete ske med rad nummer tre, d v s den nya raden. Denna rad kommer att bli en textrad. Den nya raden pekas ut i datastrukturen med index två. För att en rad ska bli av typen "Textrad" ska värdet "T" sättas i fältet `ADK_OOI_ROW_TYPE_OF_ROW`. Den text som ska stå på textraden sätts i fältet `ADK_OOI_ROW_TEXT`.

```

pTempRow = AdkGetDataRow(pRowData, 2);
error = AdkSetStr(pTempRow, ADK_OOI_ROW_TYPE_OF_ROW, "T");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkSetStr(pTempRow, ADK_OOI_ROW_TEXT, "En textrad");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Nu har rad ett och två uppdaterats och en ny rad har skapats, denna rad har fått radnummer 3. Därefter ska huvudet kopplas samman med raderna och antalet uppdateras.

```

Error = AdkSetData(pData, ADK_OOI_HEAD_ROWS, pRowData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkSetDouble(pData, ADK_OOI_HEAD_NROWS, nNewRows);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

När alla de värden som ska sättas är satta anropas funktionen `AdkUpdate()` för att uppdatera fakturan i databasen.

```

Error = AdkUpdate(pData);
if(error.lRc != ADKE_OK)

```

```

{
    CerrorHandling rErrorHandling(&error);
}

```

Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som använts, när `AdkDeleteStruct()` anropas med huvudet `pData` som inparameter kommer även det minne som allokerats till raderna att avallokeras.

```
AdkDeleteStruct(pData);
```

Uppdatera en inkommande följesedel

För att uppdatera en inkommande följesedel används funktionen `AdkUpdate()`. Det första som ska göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```

ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));

```

Därefter ska en datastruktur av önskad typ skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_DELIVERY_NOTE_HEAD` för att tala om att det är en datastruktur av typen "Följesedelshuvud" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_DELIVERY_NOTE_HEAD);
```

Med följesedelshuvudet skapat är det dags att peka ut vilken följesedel som ska uppdateras. Detta görs genom att tilldela ett värde till följesedelns nyckelfält som är `ADK_DELIVERY_NOTE_DOCUMENT_NUMBER`.

```

Error = AdkSetDouble(pData, ADK_DELIVERY_NOTE_DOCUMENT_NUMBER, 1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

I raden ovan anges att det är inkommande följesedel nummer ett (1) som ska uppdateras. Detta värde kommer att fungera som sökvärde bland följesedlarna. Finns ingen följesedel med dokumentnummer ett (1) kommer ett felmeddelande att returneras. Först kommer följesedeln att sökas upp, vissa av de värden följesedeln innehåller sedan tidigare behövs för att göra uppdateringen.

```

Error = AdkSetSortOrder(pData, eDeliveryNoteNr);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkFind(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Därefter ska de värden som ska uppdateras sättas. Om t ex följesedelsdatumet ska uppdateras kan det göras enligt följande:

```

long lDate;
error = AdkDateToLong("2005-09-15", &lDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkSetDate(pData, ADK_DELIVERY_NOTE_DATE, lDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```


För att kunna uppdatera följesedelsdatumet anropas funktion `AdkDateToLong()` som konverterar ett datum i form av en sträng till ett `long`-värde (Juliandatum), vilket kan läggas till i databasen. När konverteringen har skett anropas `AdkSetDate()` som tilldelar ett värde av typen `long` till datastrukturen `pData`, som är datastrukturen för följesedelshuvudet.

Det är även möjligt att uppdatera och skapa nya rader på en följesedel. För att kunna uppdatera en rad eller lägga till en rad måste information om antalet rader finnas. Denna information fås genom att antalet rader läsas från fältet `ADK_DELIVERY_NOTE_NROWS`. Detta är möjligt tack vare att följesedeln tidigare söktes upp med `AdkFind()`.

```
Double nRows;
error = AdkGetDouble(pData, ADK_DELIVERY_NOTE_NROWS, &nRows);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu innehåller variabeln `nRows` antalet rader. Om dessa två rader ska uppdateras behöver inga nya rader skapas. Raderna kom med datastrukturen vid `AdkFind()` ovan, och kan hämtas fram med hjälp av `AdkGetData()`. Om däremot en rad ska läggas till ska datastruktur för fler än två rader skapas. För att undvika minnesläckage måste först datastrukturen tömmas:

```
error = AdkResetStruct(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Eftersom vi har rensat datastrukturen behöver följesedelsnummer och datum sättas på samma sätt som tidigare i exemplet:

```
error = AdkSetDouble(pData, ADK_DELIVERY_NOTE_DOCUMENT_NUMBER, 1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

long lDate;
error = AdkDateToLong("2005-09-15", &lDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkSetDate(pData, ADK_DELIVERY_NOTE_DATE, lDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Det antal över två som skapas kommer att bli nya rader på följesedeln. I exemplet nedan uppdateras deläggs en rad till och en datastruktur med tre rader skapas således.

```
Int nNewRows = 3;
PADK_DATA pRowData = AdkCreateDataRow(ADK_DB_DELIVERY_NOTE_ROW, nNewRows);
```

Först uppdateras rad ett med ett nytt antal för levererade artiklar. Rad ett pekas ut med hjälp av funktionen `AdkGetDataRow()`, till funktionen skickas noll som index. Index noll motsvarar rad ett, index ett motsvarar rad två o s v. En temporär datastruktur `pTempRow` skapas för att hålla aktuell rad.

```
PADK_DATA pTempRow = AdkGetDataRow(pRowData, 0);
error = AdkSetDouble(pTempRow, ADK_OOI_ROW_QUANTITY1, 100);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Arbetet med den temporära datastrukturen `pTempRow` sker på samma sätt som arbetet med ett dokumenthuvud. I exemplet nedan ska även antalet levererade artiklar uppdateras på rad nummer två och därför pekas index ett ut och det blir möjligt att arbeta med rad nummer två.

```
pTempRow = AdkGetDataRow(pRowData, 1);
error = AdkSetDouble(pTempRow, ADK_OOI_ROW_QUANTITY1, 200);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Därefter ska arbete ske med rad nummer tre, dvs den nya raden. Denna rad kommer läsa upp en rad från en beställning. Den nya raden pekas ut i datastrukturen med index två. Värt att notera är att uppdatering av rader som redan är upplästa på följesedeln går inte att uppdatera genom att ange beställningen löpnummer. Utan uppdatering av dessa rader sker på det vis som synes i exemplet ovan.

```
pTempRow = AdkGetDataRow(pRowData, 2);

error = AdkSetStr(pTempData, ADK_OOI_ROW_FROM_TYPE, "B");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetDouble(pTempData, ADK_OOI_ROW_FROM_DOCUMENT, 11);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetDouble(pTempData, ADK_OOI_ROW_FROM_DOCROW, 2);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetDouble(pTempData, ADK_OOI_ROW_QUANTITY1, 30);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu har rad ett och två uppdaterats och en ny rad har skapats, denna rad har fått radnummer 3. Därefter ska huvudet kopplas samman med raderna och antalet uppdateras.

```
Error = AdkSetData(pData, ADK_DELIVERY_NOTE_ROWS, pRowData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkSetDouble(pData, ADK_DELIVERY_NOTE_NROWS, nNewRows);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

När alla de värden som ska sättas är satta anropas funktionen `AdkUpdate()` för att uppdatera den inkommande följesedeln i databasen.

```
Error = AdkUpdate(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som använts, när `AdkDeleteStruct()` anropas med huvudet `pData` som inparameter kommer även det minne som allokerats till raderna att avallokeras.

```
AdkDeleteStruct(pData);
```

Uppdatera ett kolli

För att uppdatera ett kolli används funktionen `AdkUpdate()`. Det första som ska göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;  
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur av önskad typ skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_PACKAGE_HEAD` för att tala om att det är en datastruktur av typen "Kolli" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_PACKAGE_HEAD);
```

Med kollit skapat är det dags att peka ut vilket kolli som ska uppdateras. Detta görs genom att tilldela värden till kollits nyckelfält som är dess kollinummer, `ADK_PACKAGE_HEAD_PACKAGE_NUMBER`, samt orderns löpnummer, `ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT`.

```
Error = AdkSetStr(pData, ADK_PACKAGE_HEAD_PACKAGE_NUMBER, "1");  
if(error.lRc != ADKE_OK)  
{  
    CerrorHandling rErrorHandling(&error);  
}  
  
error = AdkSetDouble(pData, ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT, 1);  
if(error.lRc != ADKE_OK)  
{  
    CerrorHandling rErrorHandling(&error);  
}
```

I raden ovan anges att det är kolli nummer ett (1) på order nummer ett (1) som ska uppdateras. Dessa värde kommer att fungera som sökvärden bland kollina. Finns inget kolli med kollinummer ett (1) kommer ett felmeddelande att returneras. Först kommer kollit att sökas upp, vissa av de värden kollit innehåller sedan tidigare behövs för att göra uppdateringen.

```
Error = AdkSetSortOrder(pData, ePackageOrderNr);  
if(error.lRc != ADKE_OK)  
{  
    CerrorHandling rErrorHandling(&error);  
}  
  
error = AdkFind(pData);  
if(error.lRc != ADKE_OK)  
{  
    CerrorHandling rErrorHandling(&error);  
}
```

Därefter ska de värden som ska uppdateras sättas. Om t ex bruttovikten ska uppdateras kan det göras enligt följande:

```
error = AdkSetDouble(pData, ADK_PACKAGE_HEAD_GROSS_WEIGHT, 5);  
if(error.lRc != ADKE_OK)  
{  
    CerrorHandling rErrorHandling(&error);  
}
```

För att kunna uppdatera en rad eller lägga till en rad måste information om antalet rader finnas. Denna information fås genom att antalet rader läses från fältet `ADK_PACKAGE_HEAD_NROWS`. Detta är möjligt tack vare att kollit tidigare söktes upp med `AdkFind()`.

```
Double nRows;  
error = AdkGetDouble(pData, ADK_PACKAGE_HEAD_NROWS, &nRows);  
if(error.lRc != ADKE_OK)
```

```

{
    CerrorHandling rErrorHandling(&error);
}

```

Nu innehåller variabeln `nRows` antalet rader. Om dessa två rader ska uppdateras behöver inga ny rader skapas. Raderna kom med datastrukturen vid `AdkFind()` ovan, och kan hämtas fram med hjälp av `AdkGetData()`. Om däremot en rad ska läggas till ska datastruktur för fler än två rader skapas. För att undvika minnesläckage måste först datastrukturen tömmas:

```

error = AdkResetStruct(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Eftersom vi har rensat datastrukturen behöver fälten vi behövde för att söka upp kollit sättas igen och bruttovikt sättas på samma sätt som tidigare i exemplet:

```

error = AdkSetStr(pData, ADK_PACKAGE_HEAD_PACKAGE_NUMBER, "1");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetDouble(pData, ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT, 1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetDouble(pData, ADK_PACKAGE_HEAD_GROSS_WEIGHT, 5);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Uppdatering av rader på kolli sker på samma vis som med övriga dokument, med skillnaden att orderradsnummer är ett obligatoriskt fält. Därefter ska huvudet kopplas samman med raderna och antalet uppdateras.

```

Error = AdkSetData(pData, ADK_PACKAGE_HEAD_ROWS, pRowData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkSetDouble(pData, ADK_PACKAGE_HEAD_NROWS, nNewRows);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

När alla de värden som ska sättas är satta anropas funktionen `AdkUpdate()` för att uppdatera kollit i databasen.

```

Error = AdkUpdate(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som använts, när `AdkDeleteStruct()` anropas med huvudet `pData` som inparameter kommer även det minne som allokerats till raderna att avallokeras.

```

AdkDeleteStruct(pData);

```

Grundläggande koncept för åtkomst av poster i databasen

Detta avsnitt riktar sig till nya programmerare mot vårt Api för att ge en förståelse för hur vårt Api fungerar vid arbete med poster i databasen.

För att röra sig i databasen används funktionerna `AdkFirstEx/AdkFirst`, `AdkNextEx/AdkNext`, `AdkLastEx/AdkLast`, `AdkPreviousEx/AdkPrevious` samt `AdkFind2/AdkFindEx/AdkFind`.

Grundregeln är att använda senaste varianten av respektive funktion, men ibland kan de äldre varianterna behöva användas.

Alla dessa funktioner tar som inparameter en pekare till en `ADK_DATA`-struktur. Resultatet av anropet lagras i denna och innehåller värden från en post i databasen vid ett lyckat anrop.

Apiet innehåller inget recordset utan har bara en pekare till databasen man arbetar med. Det man gör med funktionerna ovan är att flytta pekaren till en annan post i databasen och då kopieras innehållet från den posten in till den `ADK_DATA`-struktur som pekats ut vid anropet.

Det går att styra sökordning och filtrering för den databasfil man arbetar med, genom anrop till `AdkSetSortOrder`. Pga databasens grundteknik historiskt sett är sökbara fält begränsade till de sökordningar och de fält som definierats i avsnittet `AdkFind2`, `AdkFindEx`, `AdkFind` ovan.

Avsnittet om att arbeta med tidsstämplar nedan är ett exempel där man börjar att läsa från en viss post och sedan resterande poster i databasen. Notera speciellt att man börjar loopen med `AdkFind` och därefter anropar `AdkNext` tills man nått sista posten.

AdkFirst

Hämta första kunden

För att hämta första kunden utifrån aktuell sorteringsordning anropas funktionen `AdkFirst()`. Det första som måste göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;  
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_CUSTOMER` för att tala om att det är en datastruktur av typen "Kund" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_CUSTOMER);
```

För att bestämma vilken sorteringsordning som ska användas anropas funktionen `AdkSetSortOrder()`. Här anges vilken sorteringsordning som ska användas. Det är den första posten utifrån vald ordning som kommer att hämtas. Resultat av funktionen kan variera beroende på vald sorteringsordning.

```
Error = AdkSetSortOrder(pData, eCustomerName);
```

Nedan anropas `AdkFirst()` för att hämta första posten.

```
Error = AdkFirst(pData);  
if(error.lRc != ADKE_OK)  
{  
    CerrorHandling rErrorHandling(&error);  
}
```

Om värden ska hämtas från första posten används `AdkGet`-funktionerna. Här följer ett exempel där kundens namn hämtas.

```
Char* pcTemp = new char[100];
error = AdkGetStr(pData, ADK_CUSTOMER_NAME, &pcTemp, 99);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
delete [] pcTemp;
```

Variabeln `pcTemp` innehåller nu den första kundens namn. Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som används.

```
AdkDeleteStruct(pData);
```

Hämta första fakturan

För att hämta första fakturan utifrån aktuell sorteringsordning anropas funktionen `AdkFirst()`. Det första som måste göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_INVOICE_HEAD` för att tala om att det är en datastruktur av typen "Fakturahuvud" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_INVOICE_HEAD);
```

För att bestämma vilken sorteringsordning som ska användas anropas funktionen `AdkSetSortOrder()`, här anges vilken sorteringsordning som önskas. Det är den första fakturan utifrån vald ordning som kommer att hämtas. Resultat av funktionen kan variera beroende på vald sorteringsordning.

```
Error = AdkSetSortOrder(pData, eInvoiceDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Därefter anropas `AdkFirst()` för att hämta första posten.

```
Error = AdkFirst(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Datastrukturen `pData` innehåller nu den första fakturan med dess rader. För att ta reda på vilken kund som är aktuell för denna faktura hämtas kundnumret från fakturahuvudet.

```
Char* pcCustNr = new char[17];
error = AdkGetStr(pData, ADK_OOI_HEAD_CUSTOMER_NUMBER, &pcCustNr, 16);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Variabeln `pcCustNr` innehåller nu det kundnummer som är finns på fakturan. Värdet har hämtats från datastrukturen `pData` med hjälp av funktionen `AdkGetStr()`.

Fakturadatumet hämtas med hjälp av funktionen `AdkGetDate()`. Till funktionen skickas en `long` som kommer få datumet i form av ett Juliandatum.

```

Long lDate;
error = AdkGetDate(pData, ADK_OOI_HEAD_DOCUMENT_DATE1, &lDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Nu innehåller `lDate` ett värde som kan konverteras till ett datum i form av en sträng med funktionen `AdkLongToDate()`. En temporär sträng skapas för att hålla datumet.

```

Char* pcDate = new char[11];
error = AdkLongToDate(lDate, &pcDate, 11);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Variabeln `pcDate` innehåller nu datumet på formen "YYYY-MM-DD", denna sträng måste vara minst 11 tecken lång. Om datorns datumformat är annat kommer datumet att returneras på detta format.

För att arbeta med en rad används funktionen `AdkGetData()`. I funktionsanropet ska följande anges, datastrukturen för dokumentshuvudet, `pData`, fältnamnet där raden erhålls, `ADK_OOI_HEAD_ROWS`, vilket index raden har (0 för rad 1) samt en pekare till datastrukturen där raden temporärt ska hållas, `pTempRow`.

```

Char* pcArtNr = new char[17];
PADK_DATA pTempRow;
error = AdkGetData(pData, ADK_OOI_HEAD_ROWS, 0, &pTempRow);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkGetStr(pTempRow, ADK_OOI_ROW_ARTICLE_NUMBER, &pcArtNr, 16);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Nu innehåller variabeln `pcArtNr` artikelnumret till den första raden.

Ska antalet beställda artiklar läsas upp från första raden kan det ske enligt följande.

```

Double dTemp;
error = AdkGetDouble(pTempRow, ADK_OOI_ROW_QUANTITY2, &dTemp);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Variabeln `dTemp` innehåller nu antalet beställda artiklar på första raden. När alla värden som ska hämtas från datastrukturen är hämtade och inga fler operationer önskas utföras med datastrukturen återstår endast att frigöra minnet som har allokerats.

```

Error = AdkDeleteStruct(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

AdkNext

Hämta nästa kund

För att hämta nästa kund utifrån aktuell sorteringsordning anropas funktionen `AdkNext()`. För att kunna hämta nästa post måste datastrukturen som skickas in till funktionen innehålla en post. Det

finns två funktioner som kan anropas för att få en post till datastrukturen, `AdkFirst()` och `AdkFind()`. I fallet nedan används alternativet `AdkFirst()`. Det första som måste göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_CUSTOMER` för att tala om att det är en datastruktur av typen "Kund" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_CUSTOMER);
```

För att ange vilken sorteringsordning som ska användas anropas funktionen `AdkSetSortOrder()`.

```
Error = AdkSetSortOrder(pData, eCustomerName);
```

Först måste en utgångspost hämtas. Detta sker med ett anrop till `AdkFirst()` som hämtar den första posten.

```
Error = AdkFirst(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Därefter kan nästa kund hämtas utifrån aktiv sorteringsordning och aktuell kund. `AdkNext()` är på samma sätt som `AdkFirst()` beroende av vilken sorteringsordning som är aktiv. Resultat av funktionen kan variera beroende på vald sorteringsordning.

```
Error = AdkNext(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Efter anropet till `AdkNext()` finns all data som tillhör aktuell post i datastrukturen `pData`.

Om ett värde ska hämtas från aktuell post sker detta med hjälp av funktionerna `AdkGet()`. Här följer ett exempel där kundens namn hämtas.

```
Char* pcTemp = new char[100];
error = AdkGetStr(pData, ADK_CUSTOMER_NAME, &pcTemp, 99);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
delete [] pcTemp;
```

Variabeln `pcTemp` innehåller nu kundens namn. Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som används.

```
Error= AdkDeleteStruct(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Hämta nästa faktura

Funktionen `AdkNext()` hämtar nästa posten utifrån den post som är aktiv vid anropet. En post är aktiv när datastrukturen `pData` innehåller en post. För att hämta en post till datastrukturen `pData` kan någon av funktionerna `AdkFirst()` eller `AdkFind()` användas. I det exempel som följer kommer

AdkFirst() att användas. Det första som måste göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen ADK_ERROR.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur skapas. Detta görs med funktionen AdkCreateData(). Som inparameter till AdkCreateData() skickas ADK_DB_INVOICE_HEAD för att tala om att det är en datastruktur av typen "Fakturahuvud" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_INVOICE_HEAD);
```

På samma sätt som AdkFirst() är AdkNext() beroende av vilken sorteringsordning som är aktuell. Resultat av funktionen kan alltså variera beroende på vilken sorteringsordning som är aktuell för tillfället. I exemplet nedan kommer först sorteringsordning att anges och därefter kommer första posten att hämtas.

```
Error = AdkSetSortOrder(pData, eInvoiceDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkFirst(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu har första posten lästs in i datastrukturen pData, d v s första posten utifrån sorteringsordningen fakturadatum. Det är nu möjligt att hämta nästa post utifrån den som är aktiv. Nästa post hämtas med AdkNext().

```
Error = AdkNext(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Datastrukturen innehåller nu nästa post, det är möjligt att läsa värden från huvudet och raderna på samma sätt som med AdkFirst(). När all data som ska hämtas är hämtad med AdkGet-funktionerna återstår endast att frigöra minne.

```
Error= AdkDeleteStruct(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

AdkLast

Hämta sista kunden

För att hämta sista kunden utifrån aktuell sorteringsordning anropas funktionen AdkLast(). Det första som måste göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen ADK_ERROR.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur skapas. Detta görs med funktionen AdkCreateData(). Som inparameter till AdkCreateData() skickas ADK_DB_CUSTOMER för att tala om att det är en datastruktur av typen "Kund" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_CUSTOMER);
```

För att bestämma vilken sorteringsordning som ska användas anropas funktionen `AdkSetSortOrder()`. Här anges vilken sorteringsordning som ska användas. Det är den sista posten utifrån vald ordning som kommer att hämtas. Resultat av funktionen kan variera beroende på vald sorteringsordning.

```
Error = AdkSetSortOrder(pData, eCustomerName);
```

Nedan anropas `AdkLast()` för att hämta sista posten.

```
Error = AdkLast(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Om värden ska hämtas från sista posten används `AdkGet`-funktionerna. Här följer ett exempel där kundens namn hämtas.

```
Char* pcTemp = new char[100];
error = AdkGetStr(pData, ADK_CUSTOMER_NAME, &pcTemp, 99);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
delete [] pcTemp;
```

Variabeln `pcTemp` innehåller nu den sista kundens namn. Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som används.

```
AdkDeleteStruct(pData);
```

Hämta sista fakturan

För att hämta sista fakturan utifrån aktuell sorteringsordning anropas funktionen `AdkLast()`. Det första som måste göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_INVOICE_HEAD` för att tala om att det är en datastruktur av typen "Fakturahuvud" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_INVOICE_HEAD);
```

För att bestämma vilken sorteringsordning som ska användas anropas funktionen `AdkSetSortOrder()`, här anges vilken sorteringsordning som önskas. Det är den sista fakturan utifrån vald ordning som kommer att hämtas. Resultat av funktionen kan variera beroende på vald sorteringsordning.

```
Error = AdkSetSortOrder(pData, eInvoiceDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Därefter anropas `AdkLast()` för att hämta sista posten.

```
Error = AdkLast(pData);
if(error.lRc != ADKE_OK)
{
```

```

        CerrorHandling rErrorHandling(&error);
    }

```

Datastrukturen innehåller nu den sista fakturan med dess rader, det är möjligt att läsa värden från huvudet och raderna på samma sätt som med `AdkFirst()`. När all data som ska hämtas är hämtad med `AdkGet`-funktionerna återstår endast att frigöra minne.

```

Error= AdkDeleteStruct(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

AdkPrevious

Hämta föregående kund

För att hämta föregående kund utifrån aktuell sorteringsordning anropas funktionen `AdkPrevious()`. För att kunna hämta föregående post måste datastrukturen som skickas in till funktionen innehålla en post. Det finns tre funktioner som kan anropas för att få en post till datastrukturen, `AdkFirst()`, `AdkLast()` eller `AdkFind()`. I fallet nedan används alternativet `AdkLast()`. Det första som måste göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```

ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));

```

Därefter ska en datastruktur skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_CUSTOMER` för att tala om att det är en datastruktur av typen "Kund" som ska skapas.

```

PADK_DATA pData = AdkCreateData(ADK_DB_CUSTOMER);

```

För att ange vilken sorteringsordning som ska användas anropas funktionen `AdkSetSortOrder()`.

```

Error = AdkSetSortOrder(pData, eCustomerName);

```

Först måste en utgångspost hämtas. Detta sker med ett anrop till `AdkLast()` som hämtar den sista posten.

```

Error = AdkLast(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Därefter kan föregående kund hämtas utifrån aktiv sorteringsordning och aktuell kund. `AdkPrevious()` är på samma sätt som `AdkLast()` beroende av vilken sorteringsordning som är aktiv. Resultat av funktionen kan variera beroende på vald sorteringsordning.

```

Error = AdkPrevious(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Efter anropet till `AdkPrevious()` finns all data som tillhör aktuell post i datastrukturen `pData`.

Om ett värde ska hämtas från aktuell post sker detta med hjälp av funktionerna `AdkGet()`. Här följer ett exempel där kundens namn hämtas.

```

Char* pcTemp = new char[100];
error = AdkGetStr(pData, ADK_CUSTOMER_NAME, &pcTemp, 99);
if(error.lRc != ADKE_OK)

```

```

{
    CerrorHandling rErrorHandling(&error);
}
delete [] pCTemp;

```

Variabeln `pCTemp` innehåller nu kundens namn. Slutligen anropas funktionen `AdkDeleteStruct()` som avallokerar det minne som används.

```

Error= AdkDeleteStruct(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Hämta föregående faktura

Funktionen `AdkPrevious()` hämtar föregående posten utifrån den post som är aktiv vid anropet. En post är aktiv när datastrukturen `pData` innehåller en post. För att hämta en post till datastrukturen `pData` kan någon av funktionerna `AdkFirst()`, `AdkLast()` eller `AdkFind()` användas. I det exempel som följer kommer `AdkLast()` att användas. Det första som måste göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```

ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));

```

Därefter ska en datastruktur skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_INVOICE_HEAD` för att tala om att det är en datastruktur av typen "Fakturahuvud" som ska skapas.

```

PADK_DATA pData = AdkCreateData(ADK_DB_INVOICE_HEAD);

```

På samma sätt som `AdkLast()` är `AdkPrevious()` beroende av vilken sorteringsordning som är aktuell. Resultat av funktionen kan alltså variera beroende på vilken sorteringsordning som är aktuell för tillfället. I exemplet nedan kommer först sorteringsordning att anges och därefter kommer sista posten att hämtas.

```

Error = AdkSetSortOrder(pData, eInvoiceDate);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkLast(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Nu har sista posten lästs in i datastrukturen `pData`, d v s sista posten utifrån sorteringsordningen fakturadatum. Det är nu möjligt att hämta föregående post utifrån den som är aktiv. Föregående post hämtas med `AdkPrevious()`.

```

Error = AdkPrevious(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

Datastrukturen innehåller nu föregående post, det är möjligt att läsa värden från huvudet och raderna på samma sätt som med `AdkFirst()`. När all data som ska hämtas är hämtad med `AdkGet-` funktionerna återstår endast att frigöra minne.

```

Error= AdkDeleteStruct(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

```

AdkFind

Söka efter en kund

För att söka efter en post i databasen används funktionen `AdkFind()`. Sökningen utgår ifrån den sorteringsordning som för tillfället är vald för aktuell tabell. Ett sökvärde ska anges till det fält som motsvarar aktuell sorteringsordning. Om en sökning ska ske med sorteringsordningen kundnummer i tabellen kund ska ett värde anges till fältet för kundnummer i datastrukturen. Det första som måste göras är att skapa en datastruktur för felhantering. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_CUSTOMER` för att tala om att det är en datastruktur av typen "Kund" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_CUSTOMER);
```

För att ange vilken sorteringsordning (sökordning) som ska användas, anropas funktionen `AdkSetSortOrder()` med önskad sorteringsordning som inparameter. Datastrukturen `pData` innehåller information om vilken tabell sökordningen ska ändras för och skickas också med.

```
Error = AdkSetSortOrder(pData, eCustomerNr);
```

Efter att sorteringsordning har angivits ska ett sökvärde sättas i fältet `ADK_CUSTOMER_NUMBER` med hjälp av funktionen `AdkSetStr()`.

```
Error = AdkSetStr(pData, ADK_CUSTOMER_NUMBER, "Kund1");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu finns önskad sorteringsordning och sökvärdet har satts. Därefter återstår endast att anropa `AdkFind()` för att utföra sökningen.

```
Error = AdkFind(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Datastrukturen `pData` innehåller nu all den data som tillhör den första posten i databasen som motsvarat sökvärdet. Det är nu möjligt att läsa värden från denna post med funktionerna `AdkGet()`. Om det inte sker någon träff kommer ett felmeddelande att returneras, vilket kommer att tas om hand av felhanteringen.

Sökningen ovan ger träff på delsträngen "Kund1". Om det inte finns någon kund med det kundnumret, men däremot en kund med kundnr "Kund10", kommer detta resultera i ett godkänt värde. Vill man söka på exakt värde kan man förslagsvis göra en strängjämförelse själv. Koden skulle i detta fall kunna fortsätta med (i exemplet är inte hänsyn tagen till att `strcmp()` är case-sensitive och `AdkFind` är det inte):

```
char buf[17];
AdkGetStr(pData, ADK_CUSTOMER_NUMBER, &buf);
if(strcmp("Kund1", buf) != 0)
{
    //Egen felhantering
}
```

Det är endast möjligt att få en post returnerad. Om nästa post önskas måste funktionen `AdkNext()` anropas.

```
AdkDeleteStruct(pData);
```

Söka efter en faktura

Det är möjligt att söka på ett antal olika fält för varje databastabell, dessa motsvaras av en sorteringsordning. Alla sorteringsordningar med motsvarande fält finns specificerade i filen `Adk.h`. Ett värde anges till det fält som motsvarar sorteringsordningen och detta värde kommer att fungera som sökvärde. Först måste en datastruktur för felhantering skapas. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_INVOICE_HEAD` för att tala om att det är en datastruktur av typen "Fakturahuvud" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_INVOICE_HEAD);
```

Därefter ska sorteringsordning sättas.

```
Error = AdkSetSortOrder(pData, eInvoiceNr);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Den sorteringsordning som satts är fakturanummer och nu ska ett värde anges till det fält som motsvarar fakturanummer. Detta fält är `ADK_OOI_HEAD_DOCUMENT_NUMBER`. Sökning kommer att ske på värdet ett.

```
Error = AdkSetDouble(pData, ADK_OOI_HEAD_DOCUMENT_NUMBER, 1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

När sökvärde satts, kan sökning utföras med funktionen `AdkFind()`.

```
Error = AdkFind(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Till `AdkFind()` skickas datastrukturen `pData`. I datastrukturen `pData` har ett fakturanummer satts och därmed kommer detta att fungera som sökvärde. Efter sökningen innehåller datastrukturen `pData` all data från den första post som motsvarar det sökvärde som angivits. Det är nu möjligt att hämta data från dokumenthuvudet och dokumentraderna med funktionerna `AdkGet()`. Hur detta görs visas i exemplet där första fakturan hämtades.

När all data som önskades är hämtad med `AdkGet`-funktionerna återstår endast att frigöra minne.

```
Error= AdkDeleteStruct(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Hämta flera priser från en artikel

Att hämta priser från olika prislistor på artiklar kräver att man gör en sökning mellan utläsningen av priset. I exemplet nedan visas hur två olika priser från två olika prislistor läses ut.

```
Error = AdkSetStr(pData, ADK_ARTICLE_PRICE_LIST, "1");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetStr(pData, ADK_ARTICLE_NUMBER, "Artikel1");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetSortOrder(pData, eArticleNr);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkFind(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Här görs en sökning efter artikel med artikelnummer "Artikel1" och prislista "1". Lyckas sökningen på artikeln läses priset ut. Vill man hämta priset för samma artikel på prislista "2" är en ny sökning nödvändig. Exemplet nedan visar hur detta går till. Observera att artikelnummer måste sättas igen med `AdkSetStr()` för att en ny sökning ska genomföras av `AdkFind()`.

```
Double nTemp1;
error = AdkGetDouble(pData, ADK_ARTICLE_PRICE, &nTemp1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetStr(pData, ADK_ARTICLE_PRICE_LIST, «2»);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetStr(pData, ADK_ARTICLE_NUMBER, "Artikel1");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetSortOrder(pData, eArticleNr);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkFind(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Efter den andra sökningen gjorts kan man läsa ut priset på artikeln från prislista "2" på samma sätt som ovan.

```
Double nTemp2;
error = AdkGetDouble(pData, ADK_ARTICLE_PRICE, &nTemp2);
if(error.lRc != ADKE_OK)
```

```
{
    CerrorHandling rErrorHandling(&error);
}
```

AdkDelete

Radera en kund

För att radera en post i databasen används funktionen `AdkDelete()`. Till `AdkDelete()` skickas en datastruktur. I datastrukturen ska posten som ska raderas pekars ut med hjälp av databastabellens nyckel. Ett värde anges till det fält som motsvarar nyckeln för databastabellen. Först måste en datastruktur för felhantering skapas. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```

Därefter ska en datastruktur skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_CUSTOMER` för att tala om att det är en datastruktur av typen "Kund" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_CUSTOMER);
```

Därefter ska den post som ska raderas pekars ut. Detta sker genom en tilldelning till fältet som motsvarar kundnummer, `ADK_CUSTOMER_NUMBER`.

```
Error = AdkSetStr(pData, ADK_CUSTOMER_NUMBER, "Kund1");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Funktionen `AdkSetStr()` tilldelar det värde som kommer att fungera som sökvärde, hittas ingen post som motsvarar detta sökvärde kommer ett felmeddelande att returneras från `AdkDelete()`. Därefter utförs raderingen av posten med funktionen `AdkDelete()`.

```
Error = AdkDelete(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Om raderingen av någon anledning inte kan utföras kommer ett felmeddelande att returneras. I exemplet ovan raderas kunden med kundnummer "Kund1". Minnet som allokerats för `pData` kommer att avallokeras automatiskt i funktionen `AdkDelete()` om raderingen gick bra, dvs om `error.lRc` i exemplet här, är lika med `ADKE_OK`.

AdkDeleteRow

Radera en fakturarad

Det är möjligt att radera en dokumentrad som tillhör ett dokumenthuvud. Detta sker med funktionen `AdkDeleteRow()`. Det är endast möjligt att radera en rad i taget. Först måste en datastruktur för felhantering skapas. Denna datastruktur är av typen `ADK_ERROR`.

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
```


Därefter ska en datastruktur skapas. Detta görs med funktionen `AdkCreateData()`. Som inparameter till `AdkCreateData()` skickas `ADK_DB_INVOICE_HEAD` för att tala om att det är en datastruktur av typen "Fakturahuvud" som ska skapas.

```
PADK_DATA pData = AdkCreateData(ADK_DB_INVOICE_HEAD);
```

Först måste fakturan där en rad ska tas bort pekas ut. Detta sker genom att i nyckelfältet för databastabellen anges ett värde. I fallet med faktura är det fältet `ADK_OOI_HEAD_DOCUMENT_NUMBER` som ska användas. Till detta fält anges det fakturanummer där en rad ska tas bort.

```
Error = AdkSetDouble(pData, ADK_OOI_HEAD_DOCUMENT_NUMBER, 1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Värdet ett kommer att fungera som ett sökvärde när `AdkDeleteRow()` anropas, ett fel kommer att returneras om ingen post motsvarar värdet.

För att peka ut vilken rad som ska raderas anges radens nummer. Nedan kommer rad ett att raderas.

```
Error = AdkDeleteRow(pData, 1);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Till `AdkDeleteRow()` skickas pekaren till dokumenthuvudet, `pData`, och det radnummer som raden som ska tas bort har. Om radnumret till dokumentet inte är korrekt eller om raden av någon anledning inte får raderas, kommer ett felmeddelande att returneras.

När en rad raderats kommer radnumreringen att justeras och den rad som tidigare hade nummer två blir nu rad nummer ett. När raden tagits bort skall minnet som används frigöras.

```
Error= AdkDeleteStruct(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Arbeta med tidsstämplar

Loopa igenom de leverantörsfakturer som ändrats efter en viss tidpunkt

Vi skapar först en errorstruktur och datastruktur för en leverantörsfaktura:

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));
PADK_DATA pData = AdkCreateData(ADK_DB_SUPPLIER_INVOICE_HEAD);
if(pData==NULL)
{
    //Avbryt hanteringen
}
```

Vi behöver sätta sökordningen för tidsstämplar som sorterar leverantörsfakturorna i stigande ordning.

```
Error= AdkSetSortOrder(eTimeStamp);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Därefter tar vi reda på vilket fält som används för tidsstämpeln i leverantörsfakturan.

```
Int iTimeStampField;
error= AdkGetTimeStampField(pData, &iTimeStampField);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Nu förvandlar vi ett datum och tidsslag till det format det lagras i databasen. I detta exempel jobbar vi med en hårdkodad sträng.

```
Char szDateTime[] = "2010-10-01 12:00:00";
long lDateTime;
error= AdkDateTimeToLong(szDateTime, &lDateTime);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Vi sätter också detta datum som sökvärde.

```
Error= AdkSetDate(pData, iTimeStampField, lDateTime);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
```

Följande loop går igenom de poster som är ändrade fr.o.m. tidpunkten vi angav ovan.

```
For(error = AdkFind(pData); error == ADKE_OK; error = AdkNext(pData))
{
    double dLoepnr;
    error = AdkGetDouble(pData, ADK_SUP_INV_HEAD_GIVEN_NUMBER, &dLoepnr);
    if(error.lRc == ADKE_OK)
    {
        printf("%f\n", dLoepnr);
    }
}
```

Vid sökning där eTimeStamp är satt, så får man träff på den första post (kronologiskt räknat) med en tidsstämpel som är samtidigt eller senare än sökvärdet.

Arbeta med periodiseringar

Periodiseringar kan via API:et skapas för leverantörsfakturer och verifikationer. En förändring i hur man skapar periodiseringar för leverantörsfakturer infördes i version 2015.2. Det gamla sättet att skapa periodisering för leverantörsfaktura är obsolet och ska inte användas längre. Av bakåtkompatibilitetsskäl fungerar det gamla sättet, men vi rekommenderar att lösningar som bygger på denna metod uppdateras.

Man ska alltid ange information för att koppla periodiseringen till en rad i ett dokument. Startdatum för periodiseringen och antal månader som periodiseringen ska utföras behöver anges.

Vilket konto som ska användas på periodiseringen skiljer sig mellan verifikation och leverantörsfaktura (se nedan).

Om verifikationsserie är aktiverat i företaget måste verifikationsserie för periodiseringen anges.

Om man vill ändra namnet samt verifikationstexten är dessa möjliga att ange. Dessa skapas annars automatiskt.

Därutöver kan man ange resultatenhet och projekt för periodiseringen. Anger man inget så blir det blankt.

Följande exempelkod beskriver minsta möjliga antal fält som behöver anges i fallet när man har verifikationsserie aktiverat i företaget. I exemplet har hårdkodade värden använts. Felhantering bortses från i kodexemplen.

Skapa periodisering för leverantörsfaktura

På raden som ska periodiseras i leverantörsfakturan ska kostnadskontot vara angivet (tex 4010). I periodiseringen anges periodiseringskontot i fältet för konto. (Detta är en skillnad mot tidigare modell där periodiseringskontot skulle anges på leverantörsfakturaraden och kostnadskontot anges i periodiseringen i fältet för motkonto).

Observera att typen ska sättas till PERIOD_LEVFKTNY.

```
PADK_DATA pData = AdkCreateData(ADK_DB_PERIODIC_ADJUSTMENT);
if(pData)
{
    ADK_ERROR error;

    long lDate;
    error = AdkDateToLong("2015-11-30", &lDate);

    //Sätt periodiseringskonto
    error = AdkSetStr(pData, ADK_PERIODIC_ADJUSTMENT_ACCOUNT, "1710");
    //Sätt startdatum
    error = AdkSetDate(pData, ADK_PERIODIC_ADJUSTMENT_START_DATE, lDate);
    //Sätt antal månader
    error = AdkSetDouble(pData, ADK_PERIODIC_ADJUSTMENT_NUMBER_OF_MONTHS, 4);
    //Sätt verifikationsserie för periodiseringarna
    //Endast nödvändig om verifikationsserier aktiverat i företaget
    error = AdkSetStr(pData, ADK_PERIODIC_ADJUSTMENT_VER_SERIE, "G");
    //Sätt periodiseringstyp
    error = AdkSetDouble(pData, ADK_PERIODIC_ADJUSTMENT_TYPE, PERIOD_LEVFKTNY);
    //Sätt löpnr för leverantörsfakturan
    error = AdkSetDouble(pData, ADK_PERIODIC_ADJUSTMENT_CONNECTED_DOCUMENT, 500);
    //Sätt radnummer på levfakturan *)
    error = AdkSetDouble(pData, ADK_PERIODIC_ADJUSTMENT_CONNECTED_ROW_NUMBER, 2);

    //Skapa periodiseringen
    error = AdkAdd(pData);

    //Avallokera minne
    AdkDeleteStruct(pData);
}
```

**) Radnummer är inte radens index utan bör hämtas från fältet ADK_OOI_ROW_ROWNUMBER på den rad i leverantörsfakturan som ska periodiseras.*

Skapa periodisering för verifikation

På raden i verifikationen som ska periodiseras ska periodiseringskontot vara angivet (tex 1710). Kostnadskontot (tex 4010) anges i periodiseringen i fältet för motkonto.

Typen ska sättas till PERIOD_VERIF.

```
PADK_DATA pData = AdkCreateData(ADK_DB_PERIODIC_ADJUSTMENT);
if(pData)
{
    ADK_ERROR error;

    long lDate;
    error = AdkDateToLong("2015-11-30", &lDate);

    //Sätt motkonto
    error = AdkSetStr(pData, ADK_PERIODIC_ADJUSTMENT_CONTRA_ACCOUNT, "4010");
    //Sätt startdatum
    error = AdkSetDate(pData, ADK_PERIODIC_ADJUSTMENT_START_DATE, lDate);
    //Sätt antal månader
    error = AdkSetDouble(pData, ADK_PERIODIC_ADJUSTMENT_NUMBER_OF_MONTHS, 4);
    //Sätt verifikationsserie för periodiseringarna
    //Endast nödvändig om verifikationsserier aktiverat i företaget
```

```

error = AdkSetStr(pData, ADK_PERIODIC_ADJUSTMENT_VER_SERIE, "G");
//Sät periodiseringstyp
error = AdkSetDouble(pData, ADK_PERIODIC_ADJUSTMENT_TYPE, PERIOD_VERIF);
//Sätt årsid för verifikationen som ska periodiseras
error = AdkSetStr(pData, ADK_PERIODIC_ADJUSTMENT_CONNECTED_BOOKINGYEAR_ID, "F");
//Sätt verifikationsserie för verifikationen som ska periodiseras
//Endast nödvändig om verifikationsserier aktiverat i företaget
error = AdkSetStr(pData, ADK_PERIODIC_ADJUSTMENT_CONNECTED_VER_SERIE, "A");
//Sätt löpnr för verifikationen
error = AdkSetDouble(pData, ADK_PERIODIC_ADJUSTMENT_CONNECTED_DOCUMENT, 32);
//Sätt radnummer på verifikation *)
error = AdkSetDouble(pData, ADK_PERIODIC_ADJUSTMENT_CONNECTED_ROW_NUMBER, 2);

//Skapa periodiseringen
error = AdkAdd(pData);

//Avallokera minne
AdkDeleteStruct(pData);
}

```

*) Radnummer är inte radens index utan bör hämtas från fältet ADK_VERIFICATION_ROW_ROW på den rad i verifikationen som ska periodiseras.

Dynamisk meny

Från och med version 4.4 så finns möjligheten att starta externa program från Visma Administration.

I mappen gemensamma filer har en fil med namnet *DynamicMenu.xml* lagts till. Denna fil används för att lägga till val i menyalternativet *Integration* i Visma Administration. Menyn kommer endast att finnas tillgänglig för de kunder som har Visma Administration med Visma Integration. Den kommer dessutom enbart visas om det finns något valbart alternativ.

Från och med version 5.1 kan man lägga sin egen kopia lokalt på datorn i samma mapp som programmets inifil finns. Om det finns en *DynamicMenu.xml* på denna plats så kommer denna väljas istället för den fil som ligger bland gemensamma filer. Det möjliggör olika versioner av *DynamicMenu.xml* på olika datorer med olika versioner av operativsystemet, samt ev olika sökvägar till gemensamma filer resp företag.

Tre egenskaper finns för fria texter i Menu resp MenuItem. Dessa används inte i dagsläget av programmet. De kan tex användas för att identifiera egna menyer vid uppdateringar och installationer. Dessa reserveras för framtiden och kommer vid användning underlätta när flera integrationer från en eller flera partners finns i den dynamiska menyn.

Från och med version 2013.0 får alla fönster en egen integrationsknapp i Kommandofliken. Knappen fäller ut integrationsmenyn. Det går att styra ett menyalternativ så att det visas i bara ett fönster. Det görs med egenskapen ModuleId. ModuleId ska vara en sträng enligt tabellen nedan. Bara ett ModuleId kan kopplas till ett menyalternativ. Vill man ha samma menyalternativ i flera fönster behövs en egen definition för varje. Har inget ModuleId angetts visas menyalternativet i alla fönster med integrationsknapp.

Följande tre taggar med tillhörande parametrar finns att tillgå

- <Menu> – Submeny
 - Name – Namnet på menyn
 - Company – Fri text
 - App – Fri text
 - Description – Fri text
- <MenuItem>
 - Path – Sökväg till programmet
 - ModuleId – Modul som menyalternativet ska visas i
 - Parameters – Parametrar som ska skickas med vid exekveringen
 - %P – Ger programnamnet
 - %F – Ger sökväg till aktivt företag

- %G – Ger sökväg till gemensamma filer
- %R – Ger aktuellt register
- %U – Ger unikt id
- Övrigt ger exakt det som skrivs
- Company – Fri text
- App – Fri text
- Description – Fri text
- <Separator>

Exempel

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DynamicMenuSetup>
<Menu Name="Notepad">
    <MenuItem Path="C:\NOTEPAD.EXE" ModuleId="REG_ID_CUSTOMER"
        Parameters="C:\Text.txt" Company="Visma Spcs">Anteckningar</MenuItem>
</Menu>
<Separator />
<MenuItem Path="C:\Testprogram.exe" Parameters="Test %F " Description="Detta är ett
testprogram">Testprogram</MenuItem>
</DynamicMenuSetup>
```

Den första delen skapar en ny undermeny med menyvalet Anteckningar, vilket kommer starta NOTEPAD.EXE med filen Text.txt. Menyalternativet kommer att visas endast i kundfönstrets integrationsmeny.

Den andra delen skapar menyvalet Testprogram, vilket kommer starta Testprogram.exe och skicka in värdena *Test* samt *sökvägen till aktivt bolag* som parametrar till programmet. Det innebär alltså att man kan skicka in flera (valfritt antal) parametrar in till programmet som ska startas genom att separera dem med ett blanksteg. *Exempel:* Parameters=" %R %U " för att skicka med både aktuellt register och unikt id (registerpost). Parameter vi ska byta ut måste ha ett mellanslag framför och bakom. Menyalternativet kommer att visas i alla fönsters integrationsmeny.

Tabell för returvärden vid parameter %R , samt styrning av menyalternativ med ModuleId

För parametern %R är det ett heltal som skrivs in bland startparametrarna till ditt program. Dessa finns definierade i Adk.h eller som en konstant i AdkNetWrapper.dll samt AdkNet4Wrapper.dll. När ModuleId anges i DynamicMenu.xml skall det vara strängen i tabellen som anges (se exempel ovan).

ID	Returvärde	Register
REG_ID_CUSTOMER	0	Kund
REG_ID_SUPPLIER	1	Leverantör
REG_ID_OFFER	2	Offert
REG_ID_ORDER	3	Order
REG_ID_INVOICE	4	Faktura
REG_ID_SUPPLIER_INVOICE	5	Leverantörsfaktura
REG_ID_ARTICLE	6	Artikel
REG_ID_PROJEKT	7	Projekt
REG_ID_ACCOUNT	8	Konto
REG_ID_DISCOUNT_AGREEMENT	9	Kundrabatt
REG_ID_INVENTORY_ARTICLE	10	Inventering
REG_ID_MANUAL_DELIVERY_IN	11	Manuell inleverans
REG_ID_MANUAL_DELIVERY_OUT	12	Manuell utleverans
REG_ID_BOOKING	13	Beställning
REG_ID_PRICE	14	Inköpspriser
REG_ID_MEMBER	15	Medlem
REG_ID_DELIVERY_NOTE	16	Följesedel
REG_ID_CUSTOMERPAYMENT	17	Kundinbetalning
REG_ID_SUPPLIERPAYMENT	18	Leverantörsutbetalning
REG_ID_VERIFICATION	19	Verifikationer

REG ID CONTACT	20	Kontakter
REG ID AGREEMENT	21	Avtal

Felhantering

I demoexemplet har en klass skapats för att generera ett felmeddelande. Denna klass använder funktionen `AdkGetErrorText()` för att skapa ett felmeddelande och visa det på skärmen. Nedan följer en beskrivning av hur denna hantering går till.

Deklaration av klassen.

```
Class CerrorHandling
{
public:
    CerrorHandling(ADK_ERROR error);
};
```

Till konstruktorn skickas en datastruktur av typen `ADK_ERROR`, som innehåller det felmeddelande som erhöles vid funktionsanropet.

```
CerrorHandling::CerrorHandling(ADK_ERROR error)
{
```

Först skapas det två textsträngar, dels `pchTotalText` som ska innehålla det totala felmeddelandet som ska visas på skärmen och dels `pchText` som kommer att innehålla varje enskilt felmeddelande som en temporär sträng.

```
Char* pchTotalText = new char[2500];
ZeroMemory(pchTotalText, sizeof(pchTotalText));
char* pchText = new char[500];
```

Här skapas felmeddelandet. `AdkGetErrorText()` anropas fem gånger, en gång per felkod. Till varje anrop som sker till `AdkGetErrorText()` skickas följande med:

- Datastrukturen som innehåller felet
- En variabel som talar om vilken felkod som ska hämtas
- En minnesadress till en `char`-pekare där feltexten kommer att skrivas
- Längden på `char`-pekaren där felmeddelandet kommer att skrivas

```
strcat(pchTotalText, " Meddelande:\t");
AdkGetErrorText(&error, elRc, &pchText, 500);
strcat(pchTotalText, pchText);
strcat(pchTotalText, "\n Tabell:\t\t");
AdkGetErrorText(&error, elDbTable, &pchText, 500);
strcat(pchTotalText, pchText);
strcat(pchTotalText, "\n Fält:\t\t");
AdkGetErrorText(&error, elField, &pchText, 500);
strcat(pchTotalText, pchText);
strcat(pchTotalText, "\n Funktion:\t\t");
AdkGetErrorText(&error, elFunction, &pchText, 500);
strcat(pchTotalText, pchText);
strcat(pchTotalText, "\n Programdel:\t");
AdkGetErrorText(&error, elProgramPart, &pchText, 500);
strcat(pchTotalText, pchText);
```

En meddelanderuta skapas med det totala felmeddelandet som finns i strängen `pchTotalText`.

```
MessageBox(pchTotalText, "Meddelande från Error-klassen.");

delete [] pchTotalText;
delete [] pchText;
}
```

Exceptions

I och med att vi använder externa drivrutiner för databasen mot Microsoft Sql Server har risken för exceptions ökat vid exekvering av vårt API. Det är framförallt två aspekter som är viktiga:

Vi kan ta hand om exceptions som är kända av vårt system och där vi kan garantera att minneshantering inte har kraschat. I dessa områden har vi gjort förbättringar i version 2018.6. Exempel på sådana fel kan vara datum som vårt system inte kan hantera. Dessa fel omvandlas till någon felkod och i vissa fall kan mer detaljerad information om aktuellt exception återfinnas i APIAdkLogg.txt om loggning är aktiverad.

Vi kan inte ta hand om okända krascher utanför vårt system - tex i Microsofts ODBC-drivrutiner - där vi framförallt har svårt att garantera att minnesanvändning och resursanvändning inte är korrupt. Typ av fel det kan handla om kan vara slut på minne odyl. Således kommer det i vissa fall förekomma krascher som vi släpper vidare och det blir då exceptions som resultat av anropet. I dessa fall rekommenderar vi att processen avslutas om ni fångar sådana fel. Vid fortsatt körning kan vi inte garantera att något fungerar som det ska. Designen av er lösning bör vara på ett sådant sätt att data ni vill ta vara på inte försvinner vid en exception.

Utskriftsval i kunder, leverantörer och tillhörande dokument

Följande tillåtna utskriftsval finns definierade i Adk.h samt tillgängliga i AdkNetWrapper.dll resp AdkNet4Wrapper.dll:

```
typedef enum
{
    ePrinter                = 102,
    ePDF                    = 104,
    ePDFMail                = 105,
    eAutoInvoiceElectronic = 108,
    eAutoInvoicePrintA      = 109,
    eAutoInvoiceMailPDF     = 111,
}ADK_PRINT_CHOICE;
```

Dessa motsvarar de alternativ som finns i programmets comboboxar för utskriftsval.

Sambandet blanksteg och tomma värden i strängfält i databasen

Historiska skäl har gjort att vi designade en speciallösning för att hantera att man önskar jobba med tomma strängfält.

Den designen skiljer på användningen av en tom sträng ("") och användningen av en sträng som innehåller ett blanksteg (" "). Vid tom sträng tolkar API:et det som att inget ska förändras. En sträng med ett mellanslag tolkas som att man vill arbeta med ett blankt värde i databasen.

Det är främst vi två tillfällen detta kan vara till nytta:

- Ett värde i ett strängfält ska blankas ut
- Man ska uppdatera eller radera en post där nyckelvärdet är blankt

Ska tex organisationsnummer blankas ut på en kund:

```
ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));

PADK_DATA pData = AdkCreateData(ADK_DB_CUSTOMER);

error = AdkSetStr(pData, ADK_CUSTOMER_NUMBER, "Kund1");
if(error.LRc != ADKE_OK)
```

```

{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetStr(pData, ADK_CUSTOMER_ORGANISATION_NUMBER, " ");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkUpdate(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

AdkDeleteStruct(pData);

```

Ett exempel på när radering av post med blankt i nyckelvärde kan behöva göras är husavdragsposter. När man skapar en faktura med husavdrag så skapas automatiskt en husavdragspost. Saknas personnr (organisationsnummer) på kunden blir det en post för husavdrag utan personnr. Eftersom personnr ingår i nyckelvärdet kan man inte ändra personnr på en befintlig post. Strategin får bli en annan. Eftersom posten med husavdrag skapas om automatiskt kan man inte radera den och lägga till en ny post med ett korrekt personnr. Man kommer då att ha två poster för den fakturan. Det är nödvändigt att lägga till en ny husavdragspost först och därefter radera den utan personnr. I slutändan finns då en post kvar med personnr och rätt avdrag. (Man kan även uppdatera kunden med rätt personnr och radera den befintliga husavdragsposten, så skapas en ny automatiskt med ett personnr).

För att radera en fakturas husavdragspost med tomt personnr:

```

ADK_ERROR error;
ZeroMemory(&error, sizeof(ADK_ERROR));

PADK_DATA pData = AdkCreateData(ADK_DB_TAX_REDUCTION);

error = AdkSetDouble(pData, ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMBER, 12345);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

error = AdkSetStr(pData, ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER, " ");
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}
error = AdkDelete(pData);
if(error.lRc != ADKE_OK)
{
    CerrorHandling rErrorHandling(&error);
}

AdkDeleteStruct(pData);

```

Den första posten med tomt personnr raderas. Har man flera sådana rader går det tex att loopa tills raderingen misslyckas.

Samma strategi kan användas för att med AdkFind2 kunna hitta poster med blankt i nyckelfält för vald sökordning.

Utökad loggning

Loggning görs antingen genom att ange värdet 1 (TRUE) för variabel lLogOn i anrop till AdkOpenEx() resp AdkOpenEx2() eller genom att själv skapa filen APIAdkLogg.txt i katalogen Gemensamma filer. Loggning av vissa anrop kommer då att utföras för alla företag så länge filen existerar.

Fr o m version 2017.1 har en möjlighet till utökad loggning tillkommit på företagsnivå. Denna aktiveras genom att ange följande i ftg.ini i företagskatalogen:

```
[LocalLogg]
Before=1
After=1
```

Before anger att loggning görs före själva funktionen genomförs och After loggar tillståndet efter själva anropet. Vid tex kontroller av vad en integration skickat in till en funktion (tex sökvärde till AdkFind()) används Before. Är resultatet av tex en sökning det intressanta användes After.

Resultatet sparas i aktuell företagskatalog i filen LocalAdkLogg.txt.

Observera att loggning kan påverka prestanda negativt samt att Apiet saknar mekanism för att begränsa loggfilernas storlek. Därför kan det vara viktigt att bara logga när det verkligen är nödvändigt.

Arbeta med filter

Möjligheten att sätta filter infördes i version 2018.6.

Detta är ett exempel på hur man med hjälp av filter kan hitta leverantörsfaktura för specifik leverantör med angivet fakturanr från leverantören (tex för att man vill säkerställa innan man lägger till ny leverantörsfaktura att samma redan inte är inlagd i systemet). Exemplet bygger på data i det övningsbolag vi skickar med i installationen av programmet.

```
//Förutsättningar:
//-Övningsbolaget i Visma Administration har öppnats
//-Felhantering i detta exempel görs i funktionen OutputError som inte beskrivs här
//-Utskrift av resultat görs i funktionen OutputText som inte beskrivs här

ADK_ERROR error;

PADK_DATA pData = AdkCreateData(ADK_DB_SUPPLIER_INVOICE_HEAD);

if (pData)
{
    //Sätt filter på fält leverantörsnr
    error = AdkSetFilter(pData, ADK_SUP_INV_HEAD_SUPPLIER_NUMBER, "119", 0);
    if (error.lRc != ADKE_OK)
        OutputError(&error);

    //Sätt filter på fält fakturanr
    error = AdkSetFilter(pData, ADK_SUP_INV_HEAD_INVOICE_NUMBER, "SB-1248", 0);
    if (error.lRc != ADKE_OK)
        OutputError(&error);

    error = AdkFirstEx(pData, FALSE);

    //Om det finns någon post levereras ADKE_OK, annars om det inte finns någon
    post //levereras ADKE_BOF
    if (error.lRc == ADKE_OK)
        OutputText("Det finns minst en levfaktura för leverantör 119 med
fakturanummer SB-1248\r\n");
    else if (error.lRc == ADKE_BOF)
        OutputText("Det finns ingen levfaktura för leverantör 119 med fakturanummer
SB-1248\r\n");
    else
        OutputError(&error);

    //Sätt nytt filter på fält fakturanr (det gamla filtret ersätts för det fältet)
    error = AdkSetFilter(pData, ADK_SUP_INV_HEAD_INVOICE_NUMBER, "SB-99", 0);
    if (error.lRc != ADKE_OK)
        OutputError(&error);

    error = AdkFirstEx(pData, FALSE);
```

```

//Om det finns någon post levereras ADKE_OK, annars om det inte finns någon post
//levereras ADK_BOF
if (error.lRc == ADKE_OK)
    OutputText("Det finns minst en levfaktura för leverantör 119 med
    fakturanummer SB-99\r\n");
else if (error.lRc == ADKE_BOF)
    OutputText("Det finns ingen levfaktura för leverantör 119 med fakturanummer
    SB-99\r\n");
else
    OutputError(&error);

    AdkDeleteStruct(pData);
}

//Output:
//Det finns minst en levfaktura för leverantör 119 med fakturanummer SB-1248
//Det finns ingen levfaktura för leverantör 119 med fakturanummer SB-99

```

Registerbeskrivning

I fältet typ gäller följande:

C = eChar.

D = eDouble.

L = eBool.

Date = eDate.

Data = eData.

I fältet RW/R gäller följande:

RW = Läsa och skriva.

R = Endast läsa.

Artiklar/lager – ADK_DB_ARTICLE

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_ARTICLE_NUMBER	0	C	40		RW	Artikelnummer, unikt.
ADK_ARTICLE_NAME	1	C	60		RW	Artikelns benämning på det egna språket.
ADK_ARTICLE_NAME_X (4)	2	C	60		RW	Extrafält för benämning på det egna språket. Ska följa med till dokument om fältet har innehåll. I dokument läggs texten till en egen dokumentrad.
ADK_ARTICLE_SHORT_NAME (1)	3	C	16		RW	Ett valfritt kortnamn.
ADK_ARTICLE_GROUP	4	C	6		RW	Artikelgrupp. Artikelgrupper definieras i artikelgruppsregistret.
ADK_ARTICLE_ACCOUNTING_CODE	5	C	6		RW	Konteringskod. Konteringskoder definieras i artikelkonteringsregister.
ADK_ARTICLE_STOCK_GOODS	6	L	1		RW	Sann=Lagervara. Falsk=Ej lagervara, ingen hantering av lager görs för denna artikel.
ADK_ARTICLE_UNIT_CODE	7	C	4		RW	Enhetskod. Enhetskoder definieras i enhetskodsregister.
ADK_ARTICLE_SORT_ORDER (4)	8	C	8		RW	Sorteringsbegrepp. Används enbart för att styra sorteringsordningar.
ADK_ARTICLE_QUANTITY_IN_STOCK	9	D	14	4	RW/R (6)	Antal i lager. Påverkas om fältet "lagervara" är sann av alla artikelradsposter (artrad) som har flaggan "arttrans" sann och inte är makulerad.
ADK_ARTICLE_QUANTITY_RESERVED (2)	10	D	14	4	R	Reserverat antal. Uppdateras direkt under redigering av dokument.
ADK_ARTICLE_ORDER_POINT (1)	11	D	14	4	RW	Beställningspunkt.
ADK_ARTICLE_MINIMUM_ORDER (1)	12	D	14	4	RW	Minsta antal att beställa vid varje beställningstillfälle.
ADK_ARTICLE_MAXIMUM_ORDER (1)	13	D	14	4	RW	Önskad maxnivå i lagret.
ADK_ARTICLE_ORDERED_QUANTITY (1)	14	D	14	4	R	Beställt antal (inköp).
ADK_ARTICLE_TIME_OF_DELIVERY (2)	15	D	4	0	RW	Normal leveranstid från leverantör (dagar).
ADK_ARTICLE_PLACE_IN_STOCK (2)	16	C	25		RW	Lagerplats.
ADK_ARTICLE_WEIGHT (1)	17	D	9	3	RW	Vikt per enhet.
ADK_ARTICLE_VOLUME (1)	18	D	9	3	RW	Volym per enhet.
ADK_ARTICLE_SUPPLIER_NUMBER (2)	19	C	16		RW	Huvudleverantör.
ADK_ARTICLE_INVOICE_DATE	20	Date	8		R	Datum för senaste försäljning.
ADK_ARTICLE_INVENTORY_DATE (2)	21	Date	8		R	Datum för senaste inventering.
ADK_ARTICLE_INVENTORY_QUANTITY (2)	22	D	14	4	R	Antal vid senaste inventering.
ADK_ARTICLE_INVENTORY_DIFFERENCE (2)	23	D	14	4	R	Differens antal vid senaste inventering.
ADK_ARTICLE_INVENTORY_ACCUMULATE (2)	24	D	14	4	R	Differens antal ackumulerat. Uppdateras vid inventeringsjournalen.

ADK_ARTICLE_LAST_PURCHASE_PRICE (2)	25	D	13	2	R	Senaste inköpspris
ADK_ARTICLE_NOTE_OF_OUTSTANDING_ARTICLES (1)	26	L	1		RW	Sann anger att artikeln inte ska restnoteras om de inte finns för leverans.
ADK_ARTICLE_LAND_OF_ORIGIN (1)	27	C	50		RW	Ursprungsland (namn på landet).
ADK_ARTICLE_EXPORT_CODE (1)	28	C	10		RW	Kod som används på export/import handlingar.
ADK_ARTICLE_REMARK1	29	C	60		RW	Anmärkning 1. Fält för valfri användning.
ADK_ARTICLE_REMARK2	30	C	60		RW	Anmärkning 2. Fält för valfri användning.
ADK_ARTICLE_DOCUMENT_PATH	31	C	160		RW	Dokument. Sökväg till dokument.
ADK_ARTICLE_ACCUMULATED_TURNOVER_THIS_YEAR	32	D	15	2	R	Ackumulerad omsättning under året i inhemsk valuta utan moms. Uppdateras direkt av fakturor.
ADK_ARTICLE_ACCUMULATED_TURNOVER_LAST_YEAR	33	D	15	2	R	Ackumulerad omsättning föregående år.
ADK_ARTICLE_ACCUMULATED_QUANTITY_THIS_YEAR	34	D	16	4	R	Ackumulerat antal under året. Uppdateras direkt av fakturor.
ADK_ARTICLE_ACCUMULATED_QUANTITY_LAST_YEAR	35	D	16	4	R	Ackumulerat antal föregående år.
ADK_ARTICLE_CONTRIBUTION_MARGIN_THIS_YEAR	36	D	15	2	R	Täckningsbidrag i kronor (totalt). Uppdateras direkt av fakturor.
ADK_ARTICLE_CONTRIBUTION_MARGIN_LAST_YEAR	37	D	15	2	R	Täckningsbidrag i kronor föregående år (total).
ADK_ARTICLE_CLOSING_FLAG (4)	38	L	1		RW	Sann indikerar utgående artikel.
ADK_ARTICLE_ESTIMATED_PURCHASE_PRICE	39	D	13	2	RW	Kalkylerat inköpspris per enhet.
ADK_ARTICLE_ESTIMATED_CARGO_FEE (2)	40	D	13	2	RW	Kalkylerad frakt per enhet.
ADK_ARTICLE_ESTIMATED_OTHER (2)	41	D	13	2	RW	Kalkylerade övriga inkostnader per enhet
ADK_ARTICLE_STOCK_VALUE	42	D	13	2	R	Lagervärde vid senaste lagervärdering.
ADK_ARTICLE_STOCK_VALUE_INACTIVE (2)	43	L	1		R	Sann indikerar att lagerrörelse har skett sedan senaste lagervärdering och STOCK_VALUE innehåller inte lagrets dagsvärde.
ADK_ARTICLE_PRICE_LIST	44	C	8		RW	Den prislista som ska användas. Prislistor definieras i ett eget register.
ADK_ARTICLE_PRICE	45	D	12	2	RW	Pris till angiven prislista.
ADK_ARTICLE_ESTIMATED_TOTAL_PURCHASE_PRICE (2)	46	D	15	2	R	Kalkylerad inköpskostnad, det så kallade kalkylpriset. = kalkyl inköpspris + kalkyl frakt + kalkyl övrigt
ADK_ARTICLE_DISPOSABLE_IN_STOCK (2)	47	D	14	4	R	Disponibelt antal i lagret idag = i lager – restnoterat antal
ADK_ARTICLE_TURNOVER_RATE (2)	48	D	5	2	R	Omsättningshastighet föregående år.
ADK_ARTICLE_ACCUMULATE_TURNOVER_PARCEL_THIS_YEAR (1)	49	D	15	2	R	Ackumulerad omsättning i paket.
ADK_ARTICLE_ACCUMULATE_TURNOVER_TOTAL_THIS_YEAR (1)	50	D	15	2	R	Ackumulerad omsättning.
ADK_ARTICLE_ACCUMULATE_TURNOVER_PARCEL_LAST_YEAR (1)	51	D	15	2	R	Ackumulerad omsättning föregående år i paket.
ADK_ARTICLE_ACCUMULATE_TURNOVER_TOTAL_LAST_YEAR (1)	52	D	15	2	R	Ackumulerad omsättning föregående år.
ADK_ARTICLE_ACCUMULATED_QUANTITY_PARCEL_THIS_YEAR (1)	53	D	16	4	R	Ackumulerat antal i paket.
ADK_ARTICLE_ACCUMULATED_QUANTITY_TOTAL_THIS_YEAR (1)	54	D	16	4	R	Ackumulerat antal.
ADK_ARTICLE_ACCUMULATED_QUANTITY_PARCEL_LAST_YEAR (1)	55	D	16	4	R	Ackumulerat antal föregående år i paket.
ADK_ARTICLE_ACCUMULATED_QUANTITY_TOTAL_LAST_YEAR (1)	56	D	16	4	R	Ackumulerat antal föregående år.

ADK_ARTICLE_CONTRIBUTION_MARGIN_PARCEL_THIS_YEAR (1)	57	D	15	2	R	Täckningsbidrag i paket.
ADK_ARTICLE_CONTRIBUTION_MARGIN_TOTAL_THIS_YEAR (1)	58	D	15	2	R	Täckningsbidrag.
ADK_ARTICLE_CONTRIBUTION_MARGIN_PARCEL_LAST_YEAR (1)	59	D	15	2	R	Täckningsbidrag föregående år i paket.
ADK_ARTICLE_CONTRIBUTION_MARGIN_TOTAL_LAST_YEAR (1)	60	D	15	2	R	Täckningsbidrag föregående år.
ADK_ARTICLE_CONTRIBUTION_DEGREE_THIS_YEAR	61	D	6	1	R	Täckningsgrad.
ADK_ARTICLE_CONTRIBUTION_DEGREE_PARCEL_THIS_YEAR (1)	62	D	6	1	R	Täckningsgrad i paket.
ADK_ARTICLE_CONTRIBUTION_DEGREE_TOTAL_THIS_YEAR (1)	63	D	6	1	R	Täckningsgrad totalt.
ADK_ARTICLE_CONTRIBUTION_DEGREE_LAST_YEAR	64	D	6	1	R	Täckningsgrad föregående år.
ADK_ARTICLE_CONTRIBUTION_DEGREE_PARCEL_LAST_YEAR (1)	65	D	6	1	R	Täckningsgrad föregående år i paket.
ADK_ARTICLE_CONTRIBUTION_DEGREE_TOTAL_LAST_YEAR (1)	66	D	6	1	R	Täckningsgrad totalt föregående år.
ADK_ARTICLE_BAR_CODE	67	C	32		RW	Streckkod.
ADK_ARTICLE_BAR_CODE_TYPE	68	D	4	0	RW	Streckkodstyp. Tillåtna värden, se ADK_BAR_CODE_TYPE i adk.h.
ADK_ARTICLE_PARCEL (1)	69	L	1		R	Sann indikerar paketartikel.
ADK_ARTICLE_TYPE (5)	70	C	1		RW	Artikeltyp
ADK_ARTICLE_WEBSHOP (3)	71	L	1		RW	Flagga för webshopsartikel
ADK_ARTICLE_IMAGE_PATH (3)	72	C	100		R	Bildlänk OBS! Bilden kopieras inte till undermappen \Bilder i företagskatalogen som den gör i programmet
ADK_ARTICLE_LAND_OF_ORIGIN_CODE (1)	73	C	4		RW	Ursprungsland
ADK_ARTICLE_TYPE_OF_ARTICLE	74	C	1		R	Artikeltyp, anger om det är en vanlig artikel, paketartikel (P) eller strukturartikel (S).
ADK_ARTICLE_TIMESTAMP	75	Date	10		R	Tidsstämpel
ADK_ARTICLE_INACTIVE	76	L	1		RW	Artikeln inaktiv
ADK_ARTICLE_TAX_REDUCTION_TYPES_ID (4)	77	C	4		RW	Husarbetsypid
ADK_ARTICLE_NOTE	78	C	1000		RW	Anteckningar

(1) Fältet används endast i Visma Administration 2000.

(2) Fältet används endast i Visma Administration 1000/2000.

(3) Fältet används endast i Visma Administration 500/1000/2000.

(4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

(5) Fältet används endast i Visma Förening

(6) Fältet är eReadOnly i Visma Administration 1000/2000 och eReadWrite i 500 och Förening

Kunder – ADK_DB_CUSTOMER

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CUSTOMER_NUMBER	0	C	16		RW	Kundnummer, unikt.
ADK_CUSTOMER_NAME	1	C	50		RW	Namn.
ADK_CUSTOMER_SHORT_NAME (1)	2	C	16		RW	Kortnamn.
ADK_CUSTOMER_MAILING_ADDRESS	3	C	35		RW	Postadress.
ADK_CUSTOMER_MAILING_ADDRESS2	4	C	35		RW	Postadress, rad 2.
ADK_CUSTOMER_VISITING_ADDRESS	5	C	35		RW	Besöksadress.
ADK_CUSTOMER_ZIPCODE	6	C	12		RW	Postnr .

ADK_CUSTOMER_CITY	7	C	24		RW	Ortsnamn.
ADK_CUSTOMER_COUNTRY	8	C	24		RW	Land.
ADK_CUSTOMER_TELEPHONE	9	C	20		RW	Telefonnummer.
ADK_CUSTOMER_TELEPHONE2	10	C	20		RW	Alternativt telefonnummer.
ADK_CUSTOMER_FAX	11	C	20		RW	Faxnummer.
ADK_CUSTOMER_DELIVERY_NAME	12	C	50		RW	Namn för leverans. Om blankt ska innehållet i fältet NAME användas.
ADK_CUSTOMER_DELIVERY_ADDRESS	13	C	35		RW	Postadress, gata, box för leverans. Om blankt ska innehållet i fältet MAILING_ADDRESS användas.
ADK_CUSTOMER_DELIVERY_ADDRESS2	14	C	35		RW	Postadress, gata, box för leverans, rad 2. Om blankt ska innehållet i fältet MAILING_ADDRESS2 användas.
ADK_CUSTOMER_DELIVERY_VISITING_ADDRESS	15	C	35		RW	Besöksadress för leverans. Om blankt ska innehållet i fältet VISITING_ADDRESS användas.
ADK_CUSTOMER_DELIVERY_ZIPCODE	16	C	12		RW	Postnr. Om blankt ska innehållet i fältet ZIPCODE användas.
ADK_CUSTOMER_DELIVERY_CITY	17	C	24		RW	Ortsnamn för leverans. Om blankt ska innehållet i fältet CITY användas.
ADK_CUSTOMER_DELIVERY_COUNTRY	18	C	24		RW	Land för leverans. Om blankt ska innehållet i fältet COUNTRY användas.
ADK_CUSTOMER_DELIVERY_TELEPHONE	19	C	20		RW	Telefonnummer för leveransadress. Om blankt ska innehållet i fältet TELEPHONE användas.
ADK_CUSTOMER_DELIVERY_TELEPHONE2	20	C	20		RW	Alternativt telefonnummer för leveransadress. Om blankt ska innehållet i fältet TELEPHONE2 användas.
ADK_CUSTOMER_DELIVERY_FAX	21	C	20		RW	Faxnummer på leveransadress. Om blankt ska innehållet i fältet FAX användas.
ADK_CUSTOMER_REFERENCE	22	C	50		RW	Referensperson hos kund.
ADK_CUSTOMER_BGIRO	23	C	13		RW	Bankgironummer.
ADK_CUSTOMER_PGIRO	24	C	13		RW	Plusgironummer.
ADK_CUSTOMER_ORGANISATION_NUMBER	25	C	14		RW	Organisationsnummer.
ADK_CUSTOMER_CATEGORY	26	C	4		RW	Kod för kundkategori. Kundkategorier definieras i ett eget register.
ADK_CUSTOMER_DISTRICT (2)	27	C	8		RW	Kod för distrikt. Distrikt definieras i ett eget register.
ADK_CUSTOMER_SELLER (2)	28	C	10		RW	Kod för säljare. Säljare definieras i ett eget register.
ADK_CUSTOMER_CODE_OF_TERMS_OF_DELIVERY	29	C	8		RW	Kod för leveransvillkor. Leveransvillkor definieras i ett eget register.
ADK_CUSTOMER_CODE_OF_TERMS_OF_PAYMENT	30	C	8		RW	Kod för betalningsvillkor. Betalningsvillkor definieras i ett eget register.
ADK_CUSTOMER_CODE_OF_WAY_OF_DELIVERY	31	C	8		RW	Kod för leveranssätt. Leveranssätt definieras i ett eget register.
ADK_CUSTOMER_REMINDER	32	L	1		RW	Anger om kravbrev ska kunna sändas till kunden. T= kravbrev ska kunna sändas till kunden. F= sänd inte kravbrev till kunden.
ADK_CUSTOMER_DEMAND_FEE	33	L	1		RW	Anger om kravavgift ska kunna tas ut av kunden. T= kravavgift kan debiteras, F= kravavgift ska inte debiteras.
ADK_CUSTOMER_INTEREST_INVOICE	34	L	1		RW	Anger om kunden ska kunna räntefaktureras. T= ränta kan faktureras, F= ränta ska inte faktureras.
ADK_CUSTOMER_DISPATCH_FEE	35	L	1		RW	Anger om expeditionsavgift ska läggas på kundens fakturor. T= expeditionsavgift, F= ingen expeditionsavgift.
ADK_CUSTOMER_CARGO_FEE	36	L	1		RW	Anger om fraktavgift ska läggas på kundens fakturor
ADK_CUSTOMER_CODE_OF_CURRENCY (3)	37	C	4		RW	Valutakod. Valutakoder definieras i ett eget register.
ADK_CUSTOMER_CODE_OF_LANGUAGE (3)	38	C	4		RW	Språkkod. Språkkoder definieras i ett eget register.
ADK_CUSTOMER_DISCOUNT_AGREEMENT	39	C	16		RW	Rabattavtal. Rabattavtal definieras i ett eget register.
ADK_CUSTOMER_NOTE_OF_OUTSTANDING_ORDERS (1)	40	L	1		RW	Anger om restnotering inte ska kunna göras på kundens order. T= restnotera ej på denna kund, F= restnotering ska kunna göras.
ADK_CUSTOMER_EXPORT	41	L	1		RW	Anger om kunden är en exportkund. T=export, F= ej export.
ADK_CUSTOMER_LAST_INVOICE_DATE	42	Date	8		R	Senaste fakturadatum. Uppdateras i samband med att faktura skapas.

ADK_CUSTOMER_PRICE_LIST	43	C	8		RW	Den prislista som ska användas. Om rabattavtal finns har detta fält ingen funktion. Prislistor definieras i ett eget register.
ADK_CUSTOMER_INVOICE_DISCOUNT	44	D	5	1	RW	Fakturarabatt. Om rabattavtal finns har detta fält ingen funktion.
ADK_CUSTOMER_ROW_DISCOUNT	45	D	5	1	RW	Radrabatt. Om rabattavtal finns har detta fält ingen funktion.
ADK_CUSTOMER_REMARK1	46	C	60		RW	Anteckningsfält.
ADK_CUSTOMER_REMARK2	47	C	60		RW	Anteckningsfält.
ADK_CUSTOMER_CREDIT_LIMIT	48	D	15	0	RW	Kreditgräns i inhemsk valuta.
ADK_CUSTOMER_ACCUMULATE_TURNOVER_THIS_YEAR	49	D	15	2	R	Årets omsättning i inhemsk valuta utan moms. Ränta, frakt och expeditionsavgift ingår ej.
ADK_CUSTOMER_ACCUMULATE_TURNOVER_LAST_YEAR	50	D	15	2	R	Ackumulerad omsättning föregående år.
ADK_CUSTOMER_CONTRIBUTION_MARGIN_THIS_YEAR	51	D	15	2	R	Täckningsbidrag i kronor i inhemsk valuta utan moms.
ADK_CUSTOMER_CONTRIBUTION_MARGIN_LAST_YEAR	52	D	15	2	R	Täckningsbidrag i kronor föregående år.
ADK_CUSTOMER_CLAIM	53	D	14	2	R	Fordran i inhemsk valuta (inkl eventuell moms).
ADK_CUSTOMER_AVERAGE_TIME_OF_PAYMENT_THIS_YEAR	54	D	10	5	R	Genomsnittlig betaltid.
ADK_CUSTOMER_AVERAGE_TIME_OF_PAYMENT_THIS_YEAR_NO_P	55	D	9	0	R	Antal som ligger till underlag för den uträknade betaltiden i fältet ADK_CUSTOMER_AVERAGE_PAY_TIME_THIS_YEAR.
ADK_CUSTOMER_AVERAGE_TIME_OF_PAYMENT_LAST_YEAR	56	D	10	5	R	Genomsnittlig betaltid för föregående år.
ADK_CUSTOMER_AVERAGE_TIME_OF_PAYMENT_LAST_YEAR_NO_P	57	D	9	0	R	Antal som ligger till underlag för den uträknade betaltiden i fältet ADK_CUSTOMER_AVERAGE_PAY_TIME_LAST_YEAR.
ADK_CUSTOMER_SORT_ORDER (4)	58	C	8		RW	Sorteringsbegrepp. För fri användning av användaren.
ADK_CUSTOMER_VAT_NUMBER	59	C	20		RW	Momsregistreringsnummer, VAT-nr.
ADK_CUSTOMER_SUMMARY_INVOICE (1)	60	L	1		RW	Anger om samlingsfakturer ska kunna skapas för kunden. Laddas till order som normalvärde på Samlingsfakturer. T= Samlingsfakturer kunden. F= Samlingsfakturer inte kunden.
ADK_CUSTOMER_EU_CUSTOMER	61	L	1		RW	Anger om kunden är EU-kund eller ej. Laddas till faktura som normalvärde på Kvartalsredovisning EU. Sann= EU-kund. Falsk= Inte EU-kund.
ADK_CUSTOMER_EMAIL	62	C	70		RW	E-post adress.
ADK_CUSTOMER_DOCUMENT_PATH	63	C	160		RW	Sökväg till dokument.
ADK_CUSTOMER_CONTRIBUTION_DEGREE	64	D	6	1	R	Täckningsgrad.
ADK_CUSTOMER_CONTRIBUTION_DEGREE_LAST_YEAR	65	D	6	1	R	Täckningsgrad föregående år.
ADK_CUSTOMER_LASTCHANGE	66	Date	10		R	Loggar sista ändringstillfälle, avsett för intern kontroll.
ADK_CUSTOMER_DISPATCHER (3)	67	C	7		RW	Kod för speditör. Speditör definieras i ett eget register.
ADK_CUSTOMER_EANLOC (3)	68	C	20		RW	EAN-lokaliseringsnummer
ADK_CUSTOMER_DELIVERY_EANLOC (3)	69	C	20		RW	Alternativt EAN-lokaliseringsnummer för leveransadress.
ADK_CUSTOMER_CODE_OF_COUNTRY (3)	70	C	4		RW	Landskod
ADK_CUSTOMER_DELIVERY_CODE_OF_COUNTRY (3)	71	C	4		RW	Landskod för leveransadress
ADK_CUSTOMER_FIRST_NAME (5)	72	C	24		RW	Förnamn
ADK_CUSTOMER_MEMBER (5)	73	L	1		RW	Medlemsflagga
ADK_CUSTOMER_TELEPHONE3	74	C	20		RW	Alternativt telefonnummer.
ADK_CUSTOMER_DELIVERY_TELEPHONE3	75	C	20		RW	Alternativt telefonnummer.

ADK_CUSTOMER_ADDRESS_TO_BOOKING (1)	76	L	1		RW	Flagga för överföring av adress till beställning
ADK_CUSTOMER_WSCUSTOMER (3)	77	L	1		R	Flagga för om kunden är en webbshopskund
ADK_CUSTOMER_BUILDSERVICE	78	L	1		RW	Används vid byggtjänst
ADK_CUSTOMER_EDI_ORDER	79	L	1		RW	Används inte längre. Ordererkännande (EDI)
ADK_CUSTOMER_EDI_INVOICE	80	L	1		RW	Används inte längre. Kundfaktura (EDI)
ADK_CUSTOMER_EDI_CREDITINVOICE	81	L	1		RW	Används inte längre. Kreditfaktura (EDI)
ADK_CUSTOMER_EDI_PRINT_OPTION_ORDER	82	C	10		RW	Används inte längre. Utskriftsval ordererkännande (EDI)
ADK_CUSTOMER_EDI_PRINT_OPTION_INVOICE	83	C	10		RW	Används inte längre. Utskriftsval kundfaktura (EDI)
ADK_CUSTOMER_EDI_PRINT_OPTION_CREDITINVOICE	84	C	10		RW	Används inte längre. Utskriftsval kreditfaktura (EDI)
ADK_CUSTOMER_EDI_RECEIVER_ID_ORGNO	85	C	14		RW	Används inte längre. Mottagarens organisationsnummer
ADK_CUSTOMER_EDI_RECEIVER_ID_EAN	86	C	20		RW	Används inte längre. Mottagarens EAN-lokaliseringskod
ADK_CUSTOMER_EDI_RECEIVER_ID_OTHER_ID	87	C	35		RW	Mottagarens (annan) mottagarinformation
ADK_CUSTOMER_MOBILE_PHONE	88	C	20		RW	Mobiltelefonnummer
ADK_CUSTOMER_ACCEPT_COMMERCIAL_EMAIL	89	L	1		RW	Accepterar Email (reklam)
ADK_CUSTOMER_ACCEPT_COMMERCIAL_SMS	90	L	1		RW	Accepterar SMS (reklam)
ADK_CUSTOMER_NOTES	91	C	1000		RW	Anteckningar
ADK_CUSTOMER_SIGN	92	C	8		RW	Signatur (på vald referens)
ADK_CUSTOMER_IGNORE_ORGANISATION_NUMBER_CHECK	93	L	1		RW	Ignorera organisationsnummerkontroll
ADK_CUSTOMER_TIMESTAMP	94	Date	10		R	Tidsstämpel
ADK_CUSTOMER_PROFIT_CENTRE	95	C	6		RW	Resultatenhet
ADK_CUSTOMER_PROJECT (4)	96	C	10		RW	Projekt
ADK_CUSTOMER_REFOFF_TYP (1)	97	C	1		RW	Typ för kontaktuppgifter på offert (' ' = kunduppg, 'K' = kontakt, 'E' = er ref)
ADK_CUSTOMER_REFOFF_ID (1)	98	C	8		RW	Id för kontakt på offert
ADK_CUSTOMER_REFORD_TYP (2)	99	C	1		RW	Typ för kontaktuppgifter på order (' ' = kunduppg, 'K' = kontakt, 'E' = er ref)
ADK_CUSTOMER_REFORD_ID (2)	100	C	8		RW	Id för kontakt på order
ADK_CUSTOMER_REFFLJ_TYP (2)	101	C	1		RW	Typ för kontaktuppgifter på följesedel (' ' = kunduppg, 'K' = kontakt, 'E' = er ref)
ADK_CUSTOMER_REFFLJ_ID (2)	102	C	8		RW	Id för kontakt på följesedel
ADK_CUSTOMER_REFXORD_TYP (2)	103	C	1		RW	Typ för kontaktuppgifter på extra orderdok (' ' = kunduppg, 'K' = kontakt, 'E' = er ref)
ADK_CUSTOMER_REFXORD_ID (2)	104	C	8		RW	Id för kontakt på extra orderdokument
ADK_CUSTOMER_REFFKT_TYP (3)	105	C	1		RW	Typ för kontaktuppgifter på faktura (' ' = kunduppg, 'K' = kontakt, 'E' = er ref)
ADK_CUSTOMER_REFFKT_ID (3)	106	C	8		RW	Id för kontakt på faktura
ADK_CUSTOMER_REFXFKT_TYP (3)	107	C	1		RW	Typ för kontaktuppgifter på extra fakturadok (' ' = kunduppg, 'K' = kontakt, 'E' = er ref)
ADK_CUSTOMER_REFXFKT_ID (3)	108	C	8		RW	Id för kontakt på extra fakturadokument
ADK_CUSTOMER_REFKRV_TYP (3)	109	C	1		RW	Typ för kontaktuppgifter på krav (' ' = kunduppg, 'K' = kontakt, 'E' = er ref)
ADK_CUSTOMER_REFKRV_ID (3)	110	C	8		RW	Id för kontakt på krav
ADK_CUSTOMER_DELIVERY_ID	111	D	9	0	RW	Id för avvikande leveransadress
ADK_CUSTOMER_INACTIVE	112	L	1		RW	Kunden inaktiv

ADK_CUSTOMER_PRINTOPTI ON_INVOICE (7)	113	D	4	0	RW	Utskriftsval faktura
ADK_CUSTOMER_PRINTOPTI ON_INVOICE_EX (7)	114	D	4	0	RW	Utskriftsval extra fakturadokument
ADK_CUSTOMER_PRINTOPTI ON_ORDER (2) (7)	115	D	4	0	RW	Utskriftsval order
ADK_CUSTOMER_PRINTOPTI ON_ORDER_EX (2) (7)	116	D	4	0	RW	Utskriftsval extra orderdokument
ADK_CUSTOMER_PRINTOPTI ON_DELIVERYNOTE (2) (7)	117	D	4	0	RW	Utskriftsval följesedel
ADK_CUSTOMER_PRINTOPTI ON_OFFER (1) (7)	118	D	4	0	RW	Utskriftsval offert
ADK_CUSTOMER_PRINTOPTI ON_CREDITINVOICE (7)	119	D	4	0	RW	Utskriftsval kreditfaktura
ADK_CUSTOMER_PRINTOPTI ON_INTERESTINVOICE (7)	120	D	4	0	RW	Utskriftsval räntefaktura
ADK_CUSTOMER_PRINTOPTI ON_CASH_INVOICE (7)	121	D	4	0	RW	Utskriftsval kontantnota
ADK_CUSTOMER_COPY_INV OICE	122	D	4	0	RW	Antal kopior utskrift faktura
ADK_CUSTOMER_COPY_INV OICE_EX	123	D	4	0	RW	Antal kopior utskrift extra fakturadokument
ADK_CUSTOMER_COPY_CRE DITINVOICE	124	D	4	0	RW	Antal kopior utskrift kreditfaktura
ADK_CUSTOMER_COPY_INT ERESTINVOICE	125	D	4	0	RW	Antal kopior utskrift räntefaktura
ADK_CUSTOMER_COPY_OR DER (2)	126	D	4	0	RW	Antal kopior utskrift order
ADK_CUSTOMER_COPY_OR DER_EX (2)	127	D	4	0	RW	Antal kopior utskrift extra orderdokument
ADK_CUSTOMER_COPY_DEL IVERYNOTE (2)	128	D	4	0	RW	Antal kopior utskrift följesedel
ADK_CUSTOMER_COPY_OFF ER (1)	129	D	4	0	RW	Antal kopior utskrift offert
ADK_CUSTOMER_COPY_CAS H_INVOICE	130	D	4	0	RW	Antal kopior utskrift kontantnota
ADK_CUSTOMER_ELECTRON IC_REFERENCE	131	C	35		RW	Elektronisk referens
ADK_CUSTOMER_SERVICE_ PROVIDER	132	C	35		RW	Tjänsteleverantör
ADK_CUSTOMER_AI_ID	133	C	60		R	Kundens elektroniska ID
ADK_CUSTOMER_TAX_REDUC TION_HOUSING_COOPERA TIVE_IDENTITY_NUMBER (3)	134	C	14		RW	Bostadsrättsföreningen organisationsnummer
ADK_CUSTOMER_TAX_REDUC TION_DESCRIPTION_OF_PR OPERTY (3)	135	C	40		RW	Fastighetsbeteckning/Lägenhetsnummer
ADK_CUSTOMER_DEBT_ACC OUNT	136	C	6		RW	Kundfordranskonto

(1) Fältet används endast i Visma Administration 2000.

(2) Fältet används endast i Visma Administration 1000/2000.

(3) Fältet används endast i Visma Administration 500/1000/2000.

(4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

(5) Fältet används endast i Visma Förening

(7) För tillåtna värden läs avsnitt Utskriftsval i kunder, leverantörer och tillhörande dokument ovan

Leverantörer – ADK_DB_SUPPLIER

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_SUPPLIER_NUMBER	0	C	16		RW	Leverantörsnummer, unikt
ADK_SUPPLIER_NAME	1	C	50		RW	Namn.
ADK_SUPPLIER_SHORT_NA ME (1)	2	C	16		RW	Kortnamn.
ADK_SUPPLIER_MAILING_AD DRESS	3	C	35		RW	Postadress, gata, box.
ADK_SUPPLIER_MAILING ADDRESS2	4	C	35		RW	Postadress, rad 2.

ADK_SUPPLIER_VISITING_ADDRESS	5	C	35		RW	Besöksadress.
ADK_SUPPLIER_ZIPCODE	6	C	12		RW	Postnummer.
ADK_SUPPLIER_CITY	7	C	24		RW	Ortsnamn.
ADK_SUPPLIER_COUNTRY	8	C	24		RW	Land.
ADK_SUPPLIER_TELEPHONE	9	C	20		RW	Telefonnummer
ADK_SUPPLIER_TELEPHONE 2	10	C	20		RW	Alternativt telefonnummer
ADK_SUPPLIER_FAX	11	C	20		RW	Faxnummer
ADK_SUPPLIER_REFERENCE	12	C	50		RW	Referensperson hos leverantör
ADK_SUPPLIER_OUR_REFERENCE	13	C	25		RW	Vår referens på fakturorna.
ADK_SUPPLIER_ORGANISATION_NUMBER	14	C	14		RW	Organisationsnummer.
ADK_SUPPLIER_CODE_OF_CURRENCY (3)	15	C	4		RW	Valutakod.
ADK_SUPPLIER_CODE_OF_LANGUAGE (3)	16	C	4		RW	Språkkod
ADK_SUPPLIER_BGIRO	17	C	34		RW	Bankgironummer.
ADK_SUPPLIER_PGIRO	18	C	13		RW	Plusgironummer.
ADK_SUPPLIER_OUR_CUSTOMER_NUMBER	19	C	20		RW	Vårt kundnummer hos leverantör.
ADK_SUPPLIER_CREDIT_LIMIT	20	D	15	0	RW	Kreditgräns i leverantörens valuta.
ADK_SUPPLIER_DEBT	21	D	14	2	R	Summa obetalda fakturor i inhemsk valuta.
ADK_SUPPLIER_PAYMENT_SUMMENT	22	D	14	2	R	Betalningar som är avsända men som ännu inte är bekräftade i fakturans valuta.
ADK_SUPPLIER_LAST_INVOICE_DATE	23	Date	8		R	Datum för senaste faktura.
ADK_SUPPLIER_ACCUMULATED_TURNOVER_THIS_YEAR	24	D	15	2	R	Årets inköp.
ADK_SUPPLIER_ACCUMULATED_TURNOVER_LAST_YEAR	25	D	15	2	R	Föregående års inköp.
ADK_SUPPLIER_CODE_OF_TERMS_OF_PAYMENT	26	C	8		RW	Kod för betalningsvillkor.
ADK_SUPPLIER_CODE_OF_TERMS_OF_DELIVERY	27	C	8		RW	Kod för leveransvillkor.
ADK_SUPPLIER_CODE_OF_WAY_OF_DELIVERY	28	C	8		RW	Kod för leveranssätt.
ADK_SUPPLIER_REMARK1	29	C	60		RW	Anteckningsfält.
ADK_SUPPLIER_REMARK2	30	C	60		RW	Anteckningsfält.
ADK_SUPPLIER_SORT_ORDER (4)	31	C	8		RW	Sorteringsbegrepp.
ADK_SUPPLIER_CODE_OF_COUNTRY (3)	32	C	4		RW	Landskod.
ADK_SUPPLIER_CODE_OF_CENTRAL_BANK (3)	33	C	3		RW	Betalningskod.
ADK_SUPPLIER_RECIEVER_NUMBER_OF_BANK (3)	34	D	9	0	RW	Mottagarnummer.
ADK_SUPPLIER_SWIFT_ADDRESS (3)	35	C	11		RW	BIC.
ADK_SUPPLIER_CODE_OF_BANK (3)	36	C	24		RW	Bankkod.
ADK_SUPPLIER_NAME_OF_BANK (3)	37	C	32		RW	Bankens namn.
ADK_SUPPLIER_ADDRESS_OF_BANK (3)	38	C	34		RW	Bankens 129nta li.
ADK_SUPPLIER_ZIPCODE_OF_BANK (3)	39	C	10		RW	Bankens postnummer.
ADK_SUPPLIER_CITY_OF_BANK (3)	40	C	21		RW	Bankens ort.
ADK_SUPPLIER_PAYMENT_ABBROAD (3)	41	L	1		RW	Sann: Utlandsbetalning.
ADK_SUPPLIER_BG_CODE_OF_FEE_FOR_PAYMENT_ABBROAD (3)	42	D	4	0	RW	Avgiftskod bankbetalning utland. Tillåtna värden, se ADK_CODE_BG_FOREIGN_FEE i adk.h.
ADK_SUPPLIER_EMAIL	43	C	70		RW	E-postadress.
ADK_SUPPLIER_WWW	44	C	160		RW	Web-adress.
ADK_SUPPLIER_DOCUMENT	45	C	160		RW	Sökväg externt dokument.
ADK_SUPPLIER_OCR_SUPPLIER	46	L	1		RW	Sann: OCR-Leverantör

ADK_SUPPLIER_CURRENCY_ACCOUNT (3)	47	L	1		RW	Sann: Valutakonto
ADK_SUPPLIER_EANLOC	48	C	20		RW	EAN-lokaliseringsnummer
ADK_SUPPLIER_EU_PAYMENT_SEK_EUR (3)	49	L	1		RW	Markering för att betalning ska göras i SEK eller EUR till mottagare i annat EU-land
ADK_SUPPLIER_NOTES	50	C	1000		RW	Anteckningar
ADK_SUPPLIER_MOBILE_PHONE	51	C	20		RW	Mobiltelefonnummer
ADK_SUPPLIER_SIGN	52	C	8		RW	Signatur (på vald referens)
ADK_SUPPLIER_IGNORE_ORGANISATION_NUMBER_CHECK	53	L	1		RW	Ignorera kontroll av organisationsnummer
ADK_SUPPLIER_TIMESTAMP	54	Date	10		R	Tidsstämpel
ADK_SUPPLIER_PROFIT_CENTRE	55	C	6		RW	Ressultatenhet
ADK_SUPPLIER_PROJECT (4)	56	C	10		RW	Projekt
ADK_SUPPLIER_INACTIVE	57	L	1		RW	Leverantören inaktiv
ADK_SUPPLIER_AUTOGIRO	58	L	1		RW	Autogiro
ADK_SUPPLIER_PRINTOPTION_ORDER (1) (7)	59	D	4	0	RW	Utskriftsval beställning
ADK_SUPPLIER_COPY_ORDER (1)	60	D	4	0	RW	Antal kopior utskrift beställning
ADK_SUPPLIER_EDIRECTORY_ID_OTHER_ID (1)	61	C	35		RW	Referenskod
ADK_SUPPLIER_ELECTRONIC_REFERENCE (1)	62	C	35		RW	Elektronisk referens
ADK_SUPPLIER_SERVICE_PROVIDER (1)	63	C	35		RW	Tjänsteleverantör
ADK_SUPPLIER_AI_ID (1)	64	C	60		R	Leverantörens elektroniska ID
ADK_SUPPLIER_CODE_OF_BANK_COUNTRY (3)	65	C	4		RW	Bankens landskod

(1) Fältet används endast i Visma Administration 2000.

(2) Fältet används endast i Visma Administration 1000/2000.

(3) Fältet används endast i Visma Administration 500/1000/2000.

(4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

(7) För tillåtna värden läs avsnitt Utskriftsval i kunder, leverantörer och tillhörande dokument ovan

Medlemmar – ADK_DB_MEMBER

Program: Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_MEMBER_NUMBER	0	C	16		RW	Medlemsnr
ADK_MEMBER_FIRST_NAME	1	C	24		RW	Förnamn
ADK_MEMBER_LAST_NAME	2	C	50		RW	Efternamn
ADK_MEMBER_OWN_ADDRESS	3	L	1		RW	Egen adress (för medlem i familj)
ADK_MEMBER_MAILING_ADDRESS	4	C	35		RW	Adress
ADK_MEMBER_MAILING_ADDRESS2	5	C	35		RW	Adress rad 2
ADK_MEMBER_ZIPCODE	6	C	12		RW	Postnr
ADK_MEMBER_CITY	7	C	24		RW	Ort
ADK_MEMBER_COUNTRY	8	C	24		RW	Land
ADK_MEMBER_TELEPHONE	9	C	20		RW	Telefon
ADK_MEMBER_TELEPHONE2	10	C	20		RW	Telefon2
ADK_MEMBER_PERSONAL_NUMBER	11	C	14		RW	Personnr
ADK_MEMBER_SEX	12	D	4	0	RW	Kön. Tillåtna värden, se ADK_CODE_OF_SEX i adk.h.
ADK_MEMBER_EMAIL	13	C	70		RW	Epostadress
ADK_MEMBER_DOCUMENT_PATH	14	C	160		RW	Sökväg till externt dokument
ADK_MEMBER_REMARK1	15	C	60		RW	Anmärkning 1
ADK_MEMBER_REMARK2	16	C	60		RW	Anmärkning 2
ADK_MEMBER_CATEGORY	17	C	4		RW	Medlemskategori
ADK_MEMBER_FREE_CATEGORY1	18	C	4		RW	Fri kategori 1

ADK_MEMBER_FREE_CATEGORY2	19	C	4		RW	Fri kategori 2
ADK_MEMBER_FREE_CATEGORY3	20	C	4		RW	Fri kategori 3
ADK_MEMBER_FREE_CATEGORY4	21	C	4		RW	Fri kategori 4
ADK_MEMBER_FREE_CATEGORY5	22	C	4		RW	Fri kategori 5
ADK_MEMBER_FREE_CATEGORY6	23	C	4		RW	Fri kategori 6
ADK_MEMBER_FREE_CATEGORY7	24	C	4		RW	Fri kategori 7
ADK_MEMBER_FREE_CATEGORY8	25	C	4		RW	Fri kategori 8
ADK_MEMBER_FREE_CATEGORY9	26	C	4		RW	Fri kategori 9
ADK_MEMBER_FREE_CATEGORY10	27	C	4		RW	Fri kategori 10
ADK_MEMBER_SORT_ORDER	28	C	8		RW	Sorteringsbegrepp
ADK_MEMBER_BGIRO	29	C	13		RW	Bankgiro
ADK_MEMBER_PGIRO	30	C	13		RW	Plusgiro
ADK_MEMBER_CODE_OF_TERMS_OF_PAYMENT	31	C	8		RW	Betalningsvillkor
ADK_MEMBER_PRICE_LIST	32	C	8		RW	Prislista
ADK_MEMBER_INTEREST_INVOICE	33	L	1		RW	Räntefaktura
ADK_MEMBER_REMINDER	34	L	1		RW	Kravbrev
ADK_MEMBER_DEMAND_FEE	35	L	1		RW	Kravavgift
ADK_MEMBER_FAMILY_NUMBER	36	C	16		R	Familjenr (=huvudmedlemmens medlemsnr)
ADK_MEMBER_TEL3	37	C	20		RW	Telefon3
ADK_MEMBER_CODE_OF_COUNTRY	38	C	4		RW	Landskod
ADK_MEMBER_CODE_OF_TERMS_OF_DELIVERY	39	C	8		RW	Leveransvillkor
ADK_MEMBER_CODE_OF_WAY_OF_DELIVERY	40	C	8		RW	Leveranssätt
ADK_MEMBER_MOBILE_PHONE	41	C	20		RW	Mobiltelefonnummer
ADK_MEMBER_ACCEPT_COMMERCIAL_EMAIL	42	L	1		RW	Accepterar Epost (reklam)
ADK_MEMBER_ACCEPT_COMMERCIAL_SMS	43	L	1		RW	Accepterar SMS (reklam)
ADK_MEMBER_TIMESTAMP	44	Date	10		R	Tidsstämpel
ADK_MEMBER_PRINTOPTION_INVOICE (7)	45	D	4	0	RW	Utskriftsval faktura
ADK_MEMBER_PRINTOPTION_INVOICE_EX (7)	46	D	4	0	RW	Utskriftsval extra fakturadokument
ADK_MEMBER_PRINTOPTION_CREDITINVOICE (7)	47	D	4	0	RW	Utskriftsval kreditfaktura
ADK_MEMBER_PRINTOPTION_INTERESTINVOICE (7)	48	D	4	0	RW	Utskriftsval räntefaktura
ADK_MEMBER_PRINTOPTION_CASH_INVOICE (7)	49	D	4	0	RW	Utskriftsval kontantnota
ADK_MEMBER_COPY_INVOICE	50	D	4	0	RW	Antal kopior utskrift faktura
ADK_MEMBER_COPY_INVOICE_EX	51	D	4	0	RW	Antal kopior utskrift extra fakturadokument
ADK_MEMBER_COPY_CREDITINVOICE	52	D	4	0	RW	Antal kopior utskrift kreditfaktura
ADK_MEMBER_COPY_INTERESTINVOICE	53	D	4	0	RW	Antal kopior utskrift räntefaktura
ADK_MEMBER_COPY_CASH_INVOICE	54	D	4	0	RW	Antal kopior utskrift kontantnota

(7) För tillätna värden läs avsnitt Utskriftsval i kunder, leverantörer och tillhörande dokument ovan

Prislistor – ADK_DB_CODE_OF_PRICE_LIST

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn		Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_PRICE_LIST_CODE	0	C	8		RW	Prislista
ADK_CODE_OF_PRICE_LIST_TEXT	1	C	25		RW	Benämning
ADK_CODE_OF_PRICE_LIST_CURRENCY_CODE (3)	2	C	4		RW	Valutakod
ADK_CODE_OF_PRICE_LIST_LANGUAGE_CODE (3)	3	C	4		RW	Språkkod
ADK_CODE_OF_PRICE_LIST_INCLUDING_VAT	4	L	1		RW	Sann: inklusive moms
ADK_CODE_OF_PRICE_LIST_NOTE	5	C	30		RW	Anmärkning
ADK_CODE_OF_PRICE_LIST_CHANGE_DATE	6	Date	8		R	Prisomräkningsdatum
ADK_CODE_OF_PRICE_LIST_INTERVAL1	7	D	12	4/2	RW	Avrundning skall ske from detta belopp
ADK_CODE_OF_PRICE_LIST_INTERVAL2	8	D	12	4/2	RW	Nästa eventuella avrundningsintervall skall ske from detta belopp
ADK_CODE_OF_PRICE_LIST_INTERVAL3	9	D	12	4/2	RW	Nästa eventuella avrundningsintervall skall ske from detta belopp
ADK_CODE_OF_PRICE_LIST_INTERVAL4	10	D	12	4/2	RW	Nästa eventuella avrundningsintervall skall ske from detta belopp
ADK_CODE_OF_PRICE_LIST_ROUND_OFF1	11	D	12	4/2	RW	Avrundning för INTERVAL 1. 1 = heltal, 10 = tiotal osv
ADK_CODE_OF_PRICE_LIST_ROUND_OFF2	12	D	12	4/2	RW	Avrundning för INTERVAL 2. 1 = heltal, 10 = tiotal osv
ADK_CODE_OF_PRICE_LIST_ROUND_OFF3	13	D	12	4/2	RW	Avrundning för INTERVAL. 1 = heltal, 10 = tiotal osv
ADK_CODE_OF_PRICE_LIST_ROUND_OFF4	14	D	12	4/2	RW	Avrundning för INTERVAL 4. 1 = heltal, 10 = tiotal osv
ADK_CODE_OF_PRICE_LIST_TIMESTAMP	15	Date	10		R	Tidsstämpel

(3) Fältet används endast i Visma Administration 500/1000/2000.

Försäljningspriser – ADK_DB_PRICE

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_PRICE_PRICE_LIST	0	C	8		RW	Kod för prislista.
ADK_PRICE_ARTICLE_NUMBER	1	C	40		RW	Artikelnr.
ADK_PRICE_QUANTITY	2	D	12	4	RW	Stafflingsgräns (antal fr o m). Antal är 0 när det är grundpris.
ADK_PRICE_PRICE	3	D	12	4/2	RW	Pris.
ADK_PRICE_BASE	4	L	1		R	Anger om priset avser grundpris.
ADK_PRICE_PER_CENT	5	D	5	1	RW	Används tillsammans med staffling för att ange en procentssats av grundpriset. Används vid förändring av grundpriset för att räkna om stafflingspriserna. Om procentsatsen är noll görs ingen automatisk omräkning av stafflingspriset.
ADK_PRICE_DATE_OF_CHANGE	6	Date	8		R	Datum då pris har ändrats. Uppdateras varje gång fältet PRIS ändras.
ADK_PRICE_TIMESTAMP	7	Date	10		R	Tidsstämpel

Rabattavtal – ADK_DB_DISCOUNT_AGREEMENT

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_DISCOUNT_AGREEMENT_CODE	0	C	16		RW	Kod för rabattavtal, unikt.
ADK_DISCOUNT_AGREEMENT_TEXT	1	C	30		RW	Benämning på rabattavtal.

ADK_DISCOUNT_AGREEMENT_PRICE_LIST	2	C	8		RW	Den försäljningsprislista som ska användas. Prislistor definieras i ett eget register.
ADK_DISCOUNT_AGREEMENT_INVOICE_DISCOUNT	3	D	6	1	RW	Fakturarabatt i procent.
ADK_DISCOUNT_AGREEMENT_ROW_DISCOUNT	4	D	6	1	RW	Radrabatt i procent. Denna rabatt gäller de artiklar för vilka inga andra radrabatter specificerats.
ADK_DISCOUNT_AGREEMENT_TIMESTAMP	5	Date	10		R	Tidsstämpel

Rabattavtalsrader – ADK_DB_CODE_OF_CUSTOMER_DISCOUNT_ROW

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_CODE	0	C	16		RW	Kod för rabattavtal, unikt.
ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_ARTICLE_GROUP_FLAG	1	L	1		R	Sann: raden avser artikelgruppsrabatt Falsk: raden avser artikelrabatt
ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_ARTICLE_GROUP	2	C	6		RW	Artikelgrupp
ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_ARTICLE_NUMBER	3	C	40		RW	Artikelnummer
ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_PRICELIST	4	C	8		RW	Prislista
ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_ROW_DISCOUNT	5	D	6	1	RW	Radrabatt
ADK_CODE_OF_CUSTOMER_DISCOUNT_ROW_TIMESTAMP	6	Date	10		R	Tidsstämpel

Inköpspriser – ADK_DB_ARTICLE_PURCHASE_PRICE

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_ARTICLE_PURCHASE_PRICE_ARTICLE_NUMBER	0	C	40		RW	Artikelnr.
ADK_ARTICLE_PURCHASE_PRICE_SUPPLIER_NUMBER	1	C	16		RW	Leverantörsnr.
ADK_ARTICLE_PURCHASE_PRICE_QUANTITY	2	D	12	4	RW	Stafflingsgräns (fr o m). 0 om grundpris.
ADK_ARTICLE_PURCHASE_PRICE_PRICE	3	D	12	4/2	RW	Pris.
ADK_ARTICLE_PURCHASE_PRICE_SUPPLIER_ARTICLE_NUMBER	4	C	40		RW	Leverantörens artikelnr. Används bara i poster där flaggan "grund" är sann.
ADK_ARTICLE_PURCHASE_PRICE_BASE	5	L	1		R	Anger om priset avser grundpris.
ADK_ARTICLE_PURCHASE_PRICE_CHANGED	6	L	1		RW	Sann: Inpris ändrat.
ADK_ARTICLE_PURCHASE_PRICE_CHANGED_DATE	7	Date	8		R	Datum då inpris har ändrats. Uppdateras varje gång fältet PRIS ändras.
ADK_ARTICLE_PURCHASE_PRICE_PRICE_NEW	8	D	12	4/2	RW	Nypris. Används för att förbereda nya priser
ADK_ARTICLE_PURCHASE_PRICE_PRICE_PER_CENT	9	D	9	0	R	Prisändring i procent.
ADK_ARTICLE_PURCHASE_PRICE_TIMESTAMP	10	Date	10		R	Tidsstämpel

Paketartikelrader – ADK_DB_CODE_OF_ARTICLE_PARCEL

Program: Visma Administration 2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_ARTICLE_PARCEL_NR	0	C	40		R	Paketartikelnr
ADK_CODE_OF_ARTICLE_PARCEL_ARTICLE_NR	1	C	40		R	Artikelnr
ADK_CODE_OF_ARTICLE_PARCEL_TEXT	2	C	60		R	Text
ADK_CODE_OF_ARTICLE_PARCEL_TEXT2	3	C	60		R	Text2 (om TEXT är en dokumenttextkod, dokumenttexten, annars samma som TEXT)
ADK_CODE_OF_ARTICLE_PARCEL_QUANTITY	4	D	12	4	R	Antal
ADK_CODE_OF_ARTICLE_PARCEL_TOTAL	5	D	16	4	R	Total
ADK_CODE_OF_ARTICLE_PARCEL_ROWNR	6	D	9	0	R	Radnr
ADK_CODE_OF_ARTICLE_PARCEL_TYPE_OF_ROW	7	C	1		R	Radtyp (T=textrad, annars artikelrad)
ADK_CODE_OF_ARTICLE_PARCEL_WRITE_OFFER	8	L	1		R	Sann = Raden skrivs ut på offerter
ADK_CODE_OF_ARTICLE_PARCEL_WRITE_ORDER	9	L	1		R	Sann = Raden skrivs ut på order
ADK_CODE_OF_ARTICLE_PARCEL_WRITE_INVOICE	10	L	1		R	Sann = Raden skrivs ut på fakturor
ADK_CODE_OF_ARTICLE_PARCEL_WRITE_DELIVERY_NOTE	11	L	1		R	Sann = Raden skrivs ut på följesedlar
ADK_CODE_OF_ARTICLE_PARCEL_TIMESTAMP	12	Date	10		R	Tidsstämpel

Offert, order och fakturering

Offert, order och faktura använder sig av samma fält, de som börjar på ADK_OOI_HEAD. Dessa fält kan dock anta olika egenskaper beroende på vilken typ av dokument de används som, därför kommer tabellerna att redovisas separat med rätt egenskaper för respektive typ av dokument.

Artikelrader, lagertransaktioner

Offertrader, orderrader, fakturarader, avtalsrader, leverantörsfakturarader, beställningsrader och rader på inkommande följesedel använder sig av samma fält, de som börjar på ADK_OOI_ROW. Dessa fält kan dock anta olika egenskaper beroende på vilken typ av rad de används som, därför kommer tabellerna att redovisas separat med rätt egenskaper för respektive radtyp. Respektive radtyp presenteras i anslutning till motsvarande typ av dokumenthuvud.

Offertter – ADK_DB_OFFER_HEAD

Program: Visma Administration 2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_OOI_HEAD_DOCUMENT_NUMBER	0	D	15	0	RW	Dokumentnummer från löpnummerserie. Unikt. Anger offertnummer, ordernummer eller fakturanummer.
ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER1	1	D	15	0	R	Vidarekoppling till dokumentnummer i kedjan Offert – Order – Faktura.
ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER2	2	D	15	0	R	Vidarekoppling till dokumentnummer i kedjan Offert – Order – Faktura. Offert: Ordernr.
ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER3	3	D	15	0	R	Används vid order eller faktura.
ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER4	4	D	15	0	R	Används vid order eller faktura.
ADK_OOI_HEAD_DOCUMENT_NOT_DONE	5	L	1		RW	Sann indikerar att dokumentet inte är slutbehandlat (ej utskriftsklart).
ADK_OOI_HEAD_DOCUMENT_PRINTED	6	L	1		RW	Sann indikerar att dokumentet är utskrivet. Offert: Offert utskriven.

ADK_OOI_HEAD_DOCUMENT_CANCELLED	7	L	1		RW	Sann anger att dokumentet är makulerat.
ADK_OOI_HEAD_DOCUMENT_DATE1	8	Date	8		RW	Dokumentdatum. Offert: Offertdatum.
ADK_OOI_HEAD_DOCUMENT_DATE2	9	Date	8		RW	Offert: Giltighetsdatum.
ADK_OOI_HEAD_ORDER_DELIVERED	10	L	1		R	Sann anger att leverans har skett på ordern, det innebär att artiklar anses utplockade från lager.
ADK_OOI_HEAD_ORDER_PICK_PRINTED	11	L	1		R	Sann anger att plocklista är utskriven från ordern.
ADK_OOI_HEAD_ORDER_DELIVERY_NOTE_PRINTED	12	L	1		R	Sann anger att följesedel är utskriven från ordern.
ADK_OOI_HEAD_TYPE_OF_INVOICE	13	C	1		R	Fakturatyp. F = normal faktura, K = kreditfaktura, N = kontantnota, R = räntefaktura, f = manuell faktura, k = manuell kreditfaktura. Endast möjligt att skapa normalfakturer, kontantnotor och kreditfakturer.
ADK_OOI_HEAD_CUSTOMER_NUMBER	14	C	16		RW	Kundnummer.
ADK_OOI_HEAD_CUSTOMER_NAME	15	C	50		RW	Kundnamn.
ADK_OOI_HEAD_MAILING_ADDRESS1	16	C	35		RW	Postadress.
ADK_OOI_HEAD_MAILING_ADDRESS2	17	C	35		RW	Postadress, rad 2.
ADK_OOI_HEAD_ZIPCODE	18	C	12		RW	Postnr.
ADK_OOI_HEAD_CITY	19	C	24		RW	Ortsnamn.
ADK_OOI_HEAD_COUNTRY	20	C	24		RW	Land.
ADK_OOI_HEAD_DELIVERY_NAME	21	C	50		RW	Namn för leverans. Om blankt ska innehållet i fältet NAME användas.
ADK_OOI_HEAD_DELIVERY_MAILING_ADDRESS1	22	C	35		RW	Postadress, gata, box för leverans. Om blankt ska innehållet i fältet MAILING_ADDRESS1 användas.
ADK_OOI_HEAD_DELIVERY_MAILING_ADDRESS2	23	C	35		RW	Postadress, gata, box för leverans, rad 2. Om blankt ska innehållet i fältet MAILING_ADDRESS2 användas.
ADK_OOI_HEAD_DELIVERY_ZIPCODE	24	C	12		RW	Postnr. Om blankt ska innehållet i fältet ZIPCODE användas.
ADK_OOI_HEAD_DELIVERY_CITY	25	C	24		RW	Ortsnamn för leverans. Om blankt ska innehållet i fältet CITY användas.
ADK_OOI_HEAD_DELIVERY_COUNTRY	26	C	24		RW	Land för leverans. Om blankt ska innehållet i fältet COUNTRY användas.
ADK_OOI_HEAD_CUSTOMER_REFERENCE_NAME	27	C	50		RW	Referensperson hos kund.
ADK_OOI_HEAD_CUSTOMER_ORDER_NUMBER	28	C	25		R	Används vid order eller faktura.
ADK_OOI_HEAD_CUSTOMER_REFERENCE_NUMBER	29	C	16		RW	Offert: Kundens referensnummer.
ADK_OOI_HEAD_OUR_REFERENCE_NAME	30	C	25		RW	Vår referensperson. Om företagsinställning för vår ref inte är blank laddas denna hit, annars laddas namn på säljare.
ADK_OOI_HEAD_DISTRICT_CODE	31	C	8		RW	Kod för distrikt. Distrikt definieras i ett eget register.
ADK_OOI_HEAD_SELLER_CODE	32	C	10		RW	Kod för säljare. Säljare definieras i ett eget register.
ADK_OOI_HEAD_CURRENCY_CODE	33	C	4		RW	Valutakod. Valutor definieras i ett eget register.
ADK_OOI_HEAD_LANGUAGE_CODE	34	C	4		RW	Språkkod. Språk definieras i ett eget register.
ADK_OOI_HEAD_CURRENCY_RATE	35	D	12	6	RW	Valutakurs. Om valutakursen är noll ska den betraktas som ett.
ADK_OOI_HEAD_CURRENCY_UNIT	36	D	9	0	R	Valutaenhet. Om valutaenheten är noll den betraktas som ett.
ADK_OOI_HEAD_REMINDER	37	L	1		RW	Sann anger att kravbrev ska kunna sändas.
ADK_OOI_HEAD_DEMAND_FEE	38	L	1		RW	Sann anger att kravavgift ska kunna tas ut av kunden.
ADK_OOI_HEAD_INVOICE_INTEREST	39	L	1		RW	Sann anger att räntefakturering ska kunna göras.
ADK_OOI_HEAD_EXPORT	40	L	1		RW	Sann anger export.
ADK_OOI_HEAD_INCLUDING_VAT	41	L	1		RW	Sann anger att alla priser är inkl moms.
ADK_OOI_HEAD_DO_NOT_CREATE_BACKORDER	42	L	1		RW	Sann anger att restorder ej ska kunna skapas om artiklar saknas.

ADK_OOI_HEAD_LIST_OF_PR ICE_CODE	43	C	8		RW	Kod för prislista. Finns rabattavtal har detta fält ingen funktion. Prislistor definieras i ett eget register.
ADK_OOI_HEAD_INVOICE_DI SCOUNT	44	D	5	1	RW	Fakturarabatt i procent. Finns rabattavtal har detta fält ingen funktion.
ADK_OOI_HEAD_CODE_OF_D ISCOUNT	45	C	16		RW	Kod för rabattavtal. Rabattavtal definieras i ett eget register.
ADK_OOI_HEAD_CODE_OF_T ERMS_OF_PAYMENT	46	C	8		RW	Kod för betalvillkor. Betalvillkor definieras i ett eget register.
ADK_OOI_HEAD_CODE_OF_T ERMS_OF_DELIVERY	47	C	8		RW	Kod för leveransvillkor. Leveransvillkor definieras i ett eget register.
ADK_OOI_HEAD_CODE_OF_ WAY_OF_DELIVERY	48	C	8		RW	Kod för leveranssätt. Leveranssätt definieras i ett eget register.
ADK_OOI_HEAD_TEXT1	49	C	60		RW	Fri text 1.
ADK_OOI_HEAD_TEXT2	50	C	60		RW	Fri text 2.
ADK_OOI_HEAD_TEXT3	51	C	60		RW	Fri text 3.
ADK_OOI_HEAD_SET_OF_VA T1	52	D	8	5	RW	Momssats för momskod 1. Lagras i fakturan för att se den momssats som gällde när fakturan skapades.
ADK_OOI_HEAD_SET_OF_VA T2	53	D	8	5	RW	Som SET_OF_VAT1men för momskod 2.
ADK_OOI_HEAD_SET_OF_VA T3	54	D	8	5	RW	Som SET_OF_VAT1men för momskod 3.
ADK_OOI_HEAD_SET_OF_VA T4	55	D	8	5	RW	Som SET_OF_VAT1men för momskod 4
ADK_OOI_HEAD_CARGO_AM OUNT	56	D	10	2	RW	Belopp för frakt i vald valuta.
ADK_OOI_HEAD_CARGO_VA T_CODE	57	C	1		RW	Momskod för frakt.
ADK_OOI_HEAD_DISPATCH_ FEE	58	D	10	2	RW	Expeditionsavgift i vald valuta.
ADK_OOI_HEAD_VAT_CODE_ DISPATCH_FEE	59	C	1		RW	Momskod för expeditionsavgift.
ADK_OOI_HEAD_ROUND_OF F	60	D	7	2	R	Öresavrundning i vald valuta.
ADK_OOI_HEAD_TOTAL_AMO UNT	61	D	14	2	R	Totalbelopp inkl eventuell moms. Belopp i vald valuta. Är positiv även för kreditfakturor.
ADK_OOI_HEAD_BALANCE	62	D	14	2	R	Saldo. Kundfordran i vald valuta. Positivt för vanliga fakturor, negativt för kreditfakturor. Används enbart för fakturor.
ADK_OOI_HEAD_PROJECT_C ODE	63	C	10		R	Kod för projekt. Projekt definieras i ett eget register.
ADK_OOI_HEAD_PROFIT_CE NTRE	64	C	6		R	Kod för resultatenhet. Resultatenheter definieras i ett eget register.
ADK_OOI_HEAD_BASIS_OF_ VAT0	65	D	14	2	R	Momsunderlag för momskod 0.
ADK_OOI_HEAD_BASIS_OF_ VAT1	66	D	14	2	R	Momsunderlag för momskod 1.
ADK_OOI_HEAD_BASIS_OF_ VAT2	67	D	14	2	R	Momsunderlag för momskod 2.
ADK_OOI_HEAD_BASIS_OF_ VAT3	68	D	14	2	R	Momsunderlag för momskod 3.
ADK_OOI_HEAD_BASIS_OF_ VAT4	69	D	14	2	R	Momsunderlag för momskod 4.
ADK_OOI_HEAD_AMOUNT_O F_VAT1	70	D	14	2	R	Momsbelopp för momskod 1.
ADK_OOI_HEAD_AMOUNT_O F_VAT2	71	D	14	2	R	Momsbelopp för momskod 2.
ADK_OOI_HEAD_AMOUNT_O F_VAT3	72	D	14	2	R	Momsbelopp för momskod 3.
ADK_OOI_HEAD_AMOUNT_O F_VAT4	73	D	14	2	R	Momsbelopp för momskod 4.
ADK_OOI_HEAD_PENALTY_IN TEREST_RATE	74	D	6	2	RW	Dröjsmålsräntesats (procent).
ADK_OOI_HEAD_CUSTOMER _DEMAND_ACCOUNT	75	C	6		R	Kontonummer för kundfordran.
ADK_OOI_HEAD_CARGO_AC COUNT	76	C	6		R	Kontonummer för frakt.
ADK_OOI_HEAD_DISPATCH_ FEE_ACCOUNT	77	C	6		R	Kontonummer för expeditionsavgift.
ADK_OOI_HEAD_BALANCE2	78	D	14	2	R	Saldo i inhemsk valuta.

ADK_OOI_HEAD_NUMBER_OF_ORIGINAL_INVOICE	80	D	0		R	Den faktura som en kreditfaktura krediterar.
ADK_OOI_HEAD_VAT1_ACCOUNT	81	C	15	0	R	Konto för momskod 1.
ADK_OOI_HEAD_VAT2_ACCOUNT	82	C	6		R	Konto för momskod 2.
ADK_OOI_HEAD_VAT3_ACCOUNT	83	C	6		R	Konto för momskod 3.
ADK_OOI_HEAD_VAT4_ACCOUNT	84	C	6		R	Konto för momskod 4.
ADK_OOI_HEAD_ROUND_OFF_ACCOUNT	85	C	6		R	Konto för avrundning.
ADK_OOI_HEAD_DISCOUNT_ACCOUNT	86	C	6		R	Konto för givna rabatter.
ADK_OOI_HEAD_CUSTOMER_INVOICE_VAT_NUMBER	87	C	6		RW	Momsregistreringsnummer för kunden på fakturan. VAT-nr
ADK_OOI_HEAD_SUMMARY_INVOICE	88	L	1		RW	Order: Samlingsfakturerad ordern. Faktura: Fakturan är en samlingsfaktura.
ADK_OOI_HEAD_EU_ACCOUNT_FOR_QUARTERLY	89	L	1		R	Sann: Fakturan ska kvartalsredovisas till EU.
ADK_OOI_HEAD_EU_THIRD_PART_TRADE	90	L	1		R	Sann: Fakturan är ingår i trepartshandel med EU-länder.
ADK_OOI_HEAD_SAVE_TEXT	93	L	1		RW	Sann: Behåll innehållet i textraderna (TEXT1, TEXT2, TEXT3) när order eller faktura skapas från offert respektive order.
ADK_OOI_HEAD_EXTRA_DOCUMENT_PRINTED	94	L	1		R	Sann: Extra dokument utskrivet.
ADK_OOI_HEAD_BOOK_ORDER	95	L	1		R	Sann: En order har skapats från detta dokument (beställning)
ADK_OOI_HEAD_GROSS	96	D	14	2	R	Brutto.
ADK_OOI_HEAD_NET	97	D	14	2	R	Netto. Brutto efter rabatt.
ADK_OOI_HEAD_EXCLUDING_OF_VAT	98	D	14	2	R	Totalbelopp exklusive eventuell moms i fakturans valuta
ADK_OOI_HEAD_VAT_AMOUNT	99	D	14	2	R	Total momsbelopp. Summa av AMOUNT_OF_VAT1 till 3
ADK_OOI_HEAD_CONTRIBUTION_MARGIN	100	D	14	2	R	Täckningsbidrag efter fakturarabatt. Detta fält används för summering till kundens täckningsbidrag.
ADK_OOI_HEAD_VALUE	101	D	14	2	R	Ordervärdet efter fakturarabatt. Detta fält används för summering till kundens omsättning.
ADK_OOI_HEAD_TYPE_IN_TEXT	102	C	14	2	R	Returnerar långa beteckning på fakturatypen: 'Faktura', 'Kredit', 'Kontantnota', 'Ränta', 'Man faktura' eller 'Man kredit'.
ADK_OOI_HEAD_ROWS	103	Data	14	2	RW	Pekar ut tillhörande rader
ADK_OOI_HEAD_NROWS	104	D	14	2	RW	Antalet tillhörande rader
ADK_OOI_HEAD_LOCAL_REMARK	105	C	12		RW	Kommentar (tidigare benämnt Lokal anmärkning)
ADK_OOI_HEAD_CREDIT_INVOICE_COPY_NUMBER	106	D	1		RW	Faktura att kreditera vid skapande av kreditfaktura.
ADK_OOI_HEAD_CREDIT_INVOICE_COPY_WHAT	107	D	9	0	RW	Vad som ska kopieras vid skapande av kreditfaktura. Se ADK_CREDIT_INVOICE_TO_COPY för specification i adk.h.
ADK_OOI_HEAD_EANLOC	108	C	120		RW	EAN-lokaliseringsnummer. Alternativt EAN-lokaliseringsnummer för leveransadress.
ADK_OOI_HEAD_DELIVERY_EANLOC	109	C	15	0	RW	Alternativt EAN-lokaliseringsnummer för leveransadress.
ADK_OOI_HEAD_DISPATCHER	110	C	1	0	RW	Kod för speditör. Speditör definieras i ett eget register.
ADK_OOI_HEAD_CREDIT_NOTE_STORAGE_AFFECT	111	L	1		R	Sann: Kreditfakturerad påverka lagret.
ADK_OOI_HEAD_OCR_NUMBER	112	C	20		R	OCR-nr på faktura
ADK_OOI_HEAD_CODE_OF_COUNTRY	113	C	7		RW	Landskod
ADK_OOI_HEAD_DELIVERY_CODE_OF_COUNTRY	114	C	1		RW	Landskod leveransadress
ADK_OOI_HEAD_INVOICE_BASE	115	D	25		RW	Fakturaunderlagsnr
ADK_OOI_HEAD_CONTRACT_NR	116	C	4		R	Används inte i offert
ADK_OOI_HEAD_TOT_SHIPWT	117	D	14	2	R	Används inte i offert

ADK_OOI_HEAD_DOCUMENT_DATE3	118	Date	15	0	RW	Används inte i offert
ADK_OOI_HEAD_BUILDSERVICE	119	L	1		RW	Används för byggtjänst
ADK_OOI_HEAD_LAST_PAYMENT	120	Date	14	2	R	Används inte i offert
ADK_OOI_HEAD_CASH_RECEIPT	121	L	1		RW	Används inte i offert
ADK_OOI_HEAD_STOCK_SHIP_VER_DATE	122	Date	1		RW	Används inte i offert
ADK_OOI_HEAD_REDUCED_GENERAL_PENSION_FEE	123	L	1		RW	Används inte från version 4.5
ADK_OOI_HEAD_CREATED_INVCOM	124	L	1		R	Dokumentet skapat med Visma Byråstöd aktiverat. Offert: Används ej
ADK_OOI_HEAD_CREATED_SEND	125	L	1		R	Dokumentet skickat med Visma Byråstöd. Offert: Används ej
ADK_OOI_HEAD_CREATED_RECEIVED	126	L	1		R	Dokumentet hämtat med Visma Byråstöd. Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION_INCLUDED	127	L	1		RW	Innehåller husarbete Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION_MANUALLY	128	L	1		RW	Manuell fördelning (skattereduktion) Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION_TAX_BASIS	129	D	14	2	R	Underlag för skattereduktion Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION	130	D	14	2	R	Skattereduktion Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION_NUMBER_OF_PERSONS	131	D	1		R	Antal personer (fördelad skattereduktion) Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION_BUILDING_WORK	132	L	1		RW	Fakturan avser ROT-arbete Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION_HOUSING_COOPERATIVE_IDENTITY_NUMBER	133	C	14	2	RW	Bostadsrättsföreningens organisationsnummer Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION_DESCRIPTION_OF_PROPERTY	134	C	4	0	RW	Fastighetsbeteckning Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION_AMOUNT_TO_PAY	135	D	14	2	R	Belopp att betala (efter skattereduktion) Offert: Används ej
ADK_OOI_HEAD_AGREEMENT_DATE_TYPE	136	C	14		R	Offert: Används ej
ADK_OOI_HEAD_TIMESTAMP	137	Date	100		R	Tidsstämpel
ADK_OOI_HEAD_CUSTOMER_REFERENCE_SIGN	138	C	14	2	RW	Kundrefsignatur
ADK_OOI_HEAD_DELIVERY_ID	139	D	1		RW	Id för avvikande leveransadress
ADK_OOI_HEAD_ORDER_BASE	140	D	10		R	Offert: Används ej
ADK_OOI_HEAD_ORDER_BASE_TYPE	141	C	8		R	Offert: Används ej
ADK_OOI_HEAD_ELECTRONIC_REFERENCE	142	C	35	0	R	Offert: Används ej
ADK_OOI_HEAD_PRINTOPTION (7)	143	D	15	0	RW	Utskriftsval för dokumentet
ADK_OOI_HEAD_PRINTOPTION_DELIVERY_NOTE (7)	144	D	1		R	Offert: Används ej
ADK_OOI_HEAD_PRINTOPTION_EXTRA_DOCUMENT (7)	145	D	35		RW	Utskriftsval för extradokumentet
ADK_OOI_HEAD_ELECTRONIC_INVOICE_ATTACHMENT	146	C	160	0	R	Offert: Används ej
ADK_OOI_HEAD_AI_IMPORTED_XML_NAME	147	C	4	0	R	Offert: används ej
ADK_OOI_HEAD_TAX_REDUCTION_PROC	148	D	4	0	R	Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION_MAXAMOUNT	149	D	9	0	R	Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION_GREEN_TECHNOLOGY	150	L	1		R	Innehåller skattereduktion för grön teknik. Offert: Används ej
ADK_OOI_HEAD_TAX_REDUCTION_ALL_TYPES	151	L	1		R	Innehåller skattereduktion. Offert: Används ej.

(7) För tillätna värden läs avsnitt Utskriftsval i kunder, leverantörer och tillhörande dokument ovan

Offertrader – ADK_DB_OFFER_ROW

Program: Visma Administration 2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_OOI_ROW_ARTICLE_NUMBER	0	C	40		RW	Artikelnummer
ADK_OOI_ROW_SUPPLIER_ARTICLE_NUMBER	1	C	40		R	Leverantörens artikelnummer. Används ej i offerter.
ADK_OOI_ROW_QUANTITY1	2	D	12	4	R	Antalsfält. Offert: Används ej.
ADK_OOI_ROW_QUANTITY2	3	D	12	4	RW	Extra antalsfält. Offert: Offererat antal.
ADK_OOI_ROW_QUANTITY3	4	D	12	4	R	Extra antalsfält. Offert: används ej.
ADK_OOI_ROW_PRICE_EACH_CURRENT_CURRENCY	5	D	12	4/2	RW	Pris per enhet i aktuell valuta. Inklusive eller exklusive moms.
ADK_OOI_ROW_PRICE2	6	D	12	4/2	RW	Extra fält för pris. I offert, order, och faktura används detta fält för kalkylpris per enhet utan moms. Används för att visa beräknat täckningsbidrag.
ADK_OOI_ROW_DISCOUNT	7	D	10	2	RW	Rabatt i procent eller belopp. Flagga indikerar vilket. Alltid positivt.
ADK_OOI_ROW_AMOUNT_DISCOUNT_FLAG	8	L	1		RW	Sann indikerar beloppsrabatt, annars procentrabatt. Alltid positivt.
ADK_OOI_ROW_AMOUNT_CURRENT_CURRENCY	9	D	14	2	RW	Totalt belopp i aktuell valuta inklusive eller exklusive moms. Normalt positivt.
ADK_OOI_ROW_AMOUNT_DOMESTIC_CURRENCY	10	D	14	2	R	Totalt belopp i inhemsk valuta utan moms. Normalt positivt.
ADK_OOI_ROW_ACCOUNT_NUMBER	11	C	6		R	Kontonummer.
ADK_OOI_ROW_PROFIT_CENTRE	12	C	6		R	Resultatenhet. Definieras i eget register.
ADK_OOI_ROW_PROJECT	13	C	10		R	Projekt. Definieras i eget register.
ADK_OOI_ROW_TEXT	14	C	60		RW	Används för text. Exempelvis artikelbenämning i offert, order, faktura.
ADK_OOI_ROW_UNIT	15	C	4		RW	Enhetskod. Definieras i eget register.
ADK_OOI_ROW_VAT_CODE	16	C	1		RW	Momskod. Påverkas inte om inkl moms eller ej.
ADK_OOI_ROW_NOTE_OF_OUTSTANDING_ORDERS	17	L	1		R	Används i order för att indikera att differens mellan beställt och levererat inte ska restnoteras.
ADK_OOI_ROW_DATE2	18	Date	8		R	Extra datumfält. Används i order för beräknat leveransdatum och i faktura för restnoterat leveransdatum.
ADK_OOI_ROW_TYPE_OF_ROW	19	C	1		RW	Radtyp. Användning beror på typ, normalt blank. 'T' = Textrad, 'D' – deltotal (på samlingsfaktura rad med ordersumma) 'R' – Rånterad.
ADK_OOI_ROW_CONTRIBUTION_DEGREE	20	D	6	1	R	Täckningsgrad.
ADK_OOI_ROW_CONTRIBUTION_MARGIN	21	D	14	2	R	Täckningsbidrag. Blankas om noll.
ADK_OOI_ROW_PRINT	22	L	1		RW	Anger om raden ska skrivas ut (gäller offert, order eller faktura).
ADK_OOI_ROW_PRINT2	23	L	1		R	Anger om raden ska skrivas ut (gäller följesedel i en order).
	24	U	0		R	Används inte
ADK_OOI_ROW_BAR_CODE	25	C	32		RW	Streckkod.
ADK_OOI_ROW_DATE3	26	Date	8		R	Används inte i offert
ADK_OOI_ROW_ARTICLE_TYPE	27	C	1		R	Används inte i offert
ADK_OOI_ROW_FROM_TYPE	28	C	1		R	Används inte i offert
ADK_OOI_ROW_FROM_DOCUMENT	29	D	15	0	R	Används inte i offert
ADK_OOI_ROW_FROM_DOCUMENT	30	D	9	0	R	Används inte i offert

ADK_OOI_ROW_ROWNUMBE R	31	D	9	0	R	Radnummer
ADK_OOI_ROW_CONNECTIO N_TYPE	32	C	3		R	Kopplingstyp
ADK_OOI_ROW_CONNECTIO N_DOCUMENT	33	D	15	0	R	Kopplingsdokument
ADK_OOI_ROW_CONNECTIO N_DOCROW	34	D	9	0	R	Kopplingsnummer
ADK_OOI_ROW_SERIAL_NUM BER	35	C	30		R	Används inte i offert
ADK_OOI_ROW_BATCH_NUM BER	36	C	30		R	Används inte i offert
ADK_OOI_ROW_BEST_BEFOR E	37	Date	8		R	Används inte i offert
ADK_OOI_ROW_NOTIFY_QT1	38	D	12	4	R	Används inte i offert
ADK_OOI_ROW_NOTIFY_QT2	39	D	12	4	R	Används inte i offert
ADK_OOI_ROW_PIN_BOOKIN G	40	L	1		R	Används inte i offert
ADK_OOI_ROW_NUM_PIN_BO OKING	41	D	9	0	R	Används inte i offert
ADK_OOI_ROW_LABOUR_CO ST	42	L	1		RW	Används inte i offert
ADK_OOI_ROW_QT_BOOKING	43	D	12	3	R	Kvantitet (används av leverantörsfaktura)
ADK_OOI_ROW_ORDER_BAS E_TYPE	44	C	1		R	Används inte i offert
ADK_OOI_ROW_USER_ID	45	C	15		R	Medarbetarid
ADK_OOI_ROW_STRUCTURE _ORDER	46	D	9	0	R	Ordning i struktur
ADK_OOI_ROW_PACKAGE	47	L	1		R	Paket
ADK_OOI_ROW_STRUCTURE	48	L	1		R	Struktur
ADK_OOI_ROW_TIMESTAMP	49	Date	10		R	Tidsstämpel
ADK_OOI_ROW_TAX_REDU TION_TYPES_ID	50	C	4		R	Används inte i offert.
ADK_OOI_ROW_HOURS_WOR KED	51	D	7	1	R	Används inte i offert.
ADK_OOI_ROW_H_SCHAB_P ROC	52	D	6	2	RW	Schablonprocentsats för vissa husarbetstyper och grön teknik.

Order – ADK_DB_ORDER_HEAD

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_OOI_HEAD_DOCUMENT _NUMBER	0	D	15	0	RW	Dokumentnummer från löpnummerserie. Unikt. Anger offertnummer, ordernummer eller fakturanummer.
ADK_OOI_HEAD_DOCUMENT _CONNECTION_NUMBER1	1	D	15	0	RW/R (6)	Vidarekoppling till dokumentnummer i kedjan Offert – Order – Faktura. Order: Offertnummer. 0 om direktorder
ADK_OOI_HEAD_DOCUMENT _CONNECTION_NUMBER2	2	D	15	0	R	Vidarekoppling till dokumentnummer i kedjan Offert – Order – Faktura. Order: Fakturanr.
ADK_OOI_HEAD_DOCUMENT _CONNECTION_NUMBER3	3	D	15	0	R	Order: Restordernummer om restorder har skapats.
ADK_OOI_HEAD_DOCUMENT _CONNECTION_NUMBER4	4	D	15	0	R	Order: På en restorder: Ursprungsorder; numret på den order som är upphovet till denna restorder.
ADK_OOI_HEAD_DOCUMENT _NOT_DONE	5	L	1		RW	Sann indikerar att dokumentet inte är slutbehandlat (ej utskriftsklart).
ADK_OOI_HEAD_DOCUMENT _PRINTED	6	L	1		RW	Sann indikerar att dokumentet är utskrivet. Order: Ordererkännande utskrivet.
ADK_OOI_HEAD_DOCUMENT _CANCELLED	7	L	1		RW	Sann anger att dokumentet är makulerat.
ADK_OOI_HEAD_DOCUMENT _DATE1	8	Date	8		RW	Dokumentdatum. Order: Orderdatum.
ADK_OOI_HEAD_DOCUMENT _DATE2	9	Date	8		RW	Order: Leveransdatum.
ADK_OOI_HEAD_ORDER_DEL IVERED	10	L	1		RW	Sann anger att leverans har skett på ordern, det innebär att artiklar anses utplockade från lager.
ADK_OOI_HEAD_ORDER_PIC K_PRINTED	11	L	1		RW	Sann anger att plocklista är utskriven från ordern.

ADK_OOI_HEAD_ORDER_DELIVERY_NOTE_PRINTED	12	L	1		RW	Sann anger att följesedel är utskriven från ordern.
ADK_OOI_HEAD_TYPE_OF_INVOICE	13	C	1		R	Fakturatyp. F = normal faktura, K = kreditfaktura, N = kontantnota, R = räntefaktura, f = manuell faktura, k = manuell kreditfaktura. Endast möjligt att skapa normalfakturer, kontantnotor och kreditfakturer.
ADK_OOI_HEAD_CUSTOMER_NUMBER	14	C	16		RW	Kundnummer.
ADK_OOI_HEAD_CUSTOMER_NAME	15	C	50		RW	Kundnamn.
ADK_OOI_HEAD_MAILING_ADDRESS1	16	C	35		RW	Postadress.
ADK_OOI_HEAD_MAILING_ADDRESS2	17	C	35		RW	Postadress, rad 2.
ADK_OOI_HEAD_ZIPCODE	18	C	12		RW	Postnr.
ADK_OOI_HEAD_CITY	19	C	24		RW	Ortsnamn.
ADK_OOI_HEAD_COUNTRY	20	C	24		RW	Land.
ADK_OOI_HEAD_DELIVERY_NAME	21	C	50		RW	Namn för leverans. Om blankt ska innehållet i fältet NAME användas.
ADK_OOI_HEAD_DELIVERY_MAILING_ADDRESS1	22	C	35		RW	Postadress, gata, box för leverans. Om blankt ska innehållet i fältet MAILING_ADDRESS1 användas.
ADK_OOI_HEAD_DELIVERY_MAILING_ADDRESS2	23	C	35		RW	Postadress, gata, box för leverans, rad 2. Om blankt ska innehållet i fältet MAILING_ADDRESS2 användas.
ADK_OOI_HEAD_DELIVERY_ZIPCODE	24	C	12		RW	Postnr. Om blankt ska innehållet i fältet ZIPCODE användas.
ADK_OOI_HEAD_DELIVERY_CITY	25	C	24		RW	Ortsnamn för leverans. Om blankt ska innehållet i fältet CITY användas.
ADK_OOI_HEAD_DELIVERY_COUNTRY	26	C	24		RW	Land för leverans. Om blankt ska innehållet i fältet COUNTRY användas.
ADK_OOI_HEAD_CUSTOMER_REFERENCE_NAME	27	C	50		RW	Referensperson hos kund.
ADK_OOI_HEAD_CUSTOMER_ORDER_NUMBER	28	C	25		RW	Order: Kundens ordernr.
ADK_OOI_HEAD_CUSTOMER_REFERENCE_NUMBER (1)	29	C	16		R	Används vid offert.
ADK_OOI_HEAD_OUR_REFERENCE_NAME	30	C	25		RW	Vår referensperson. Om företagsinställning för vår ref inte är blank laddas denna hit, annars laddas namn på säljare.
ADK_OOI_HEAD_DISTRICT_CODE	31	C	8		RW	Kod för distrikt. Distrikt definieras i ett eget register.
ADK_OOI_HEAD_SELLER_CODE	32	C	10		RW	Kod för säljare. Säljare definieras i ett eget register.
ADK_OOI_HEAD_CURRENCY_CODE	33	C	4		RW	Valutakod. Valutor definieras i ett eget register.
ADK_OOI_HEAD_LANGUAGE_CODE	34	C	4		RW	Språkkod. Språk definieras i ett eget register.
ADK_OOI_HEAD_CURRENCY_RATE	35	D	12	6	RW	Valutakurs. Om valutakursen är noll ska den betraktas som ett.
ADK_OOI_HEAD_CURRENCY_UNIT	36	D	9	0	R	Valutaenhet. Om valutaenheten är noll den betraktas som ett.
ADK_OOI_HEAD_REMINDER	37	L	1		RW	Sann anger att kravbrev ska kunna sändas.
ADK_OOI_HEAD_DEMAND_FEE	38	L	1		RW	Sann anger att kravavgift ska kunna tas ut av kunden.
ADK_OOI_HEAD_INVOICE_INTEREST	39	L	1		RW	Sann anger att räntefakturering ska kunna göras.
ADK_OOI_HEAD_EXPORT	40	L	1		RW	Sann anger export.
ADK_OOI_HEAD_INCLUDING_VAT	41	L	1		RW	Sann anger att alla priser är inkl moms.
ADK_OOI_HEAD_DO_NOT_CREATE_BACKORDER (1)	42	L	1		RW	Sann anger att restorder ej ska kunna skapas om artiklar saknas.
ADK_OOI_HEAD_LIST_OF_PRICE_CODE	43	C	8		RW	Kod för prislista. Finns rabattavtal har detta fält ingen funktion. Prislistor definieras i ett eget register.
ADK_OOI_HEAD_INVOICE_DISCOUNT	44	D	5	1	RW	Fakturarabatt i procent. Finns rabattavtal har detta fält ingen funktion.
ADK_OOI_HEAD_CODE_OF_DISCOUNT	45	C	16		RW	Kod för rabattavtal. Rabattavtal definieras i ett eget register.
ADK_OOI_HEAD_CODE_OF_TERMS_OF_PAYMENT	46	C	8		RW	Kod för betalvillkor. Betalvillkor definieras i ett eget register.
ADK_OOI_HEAD_CODE_OF_TERMS_OF_DELIVERY	47	C	8		RW	Kod för leveransvillkor. Leveransvillkor definieras i ett eget register.

ADK_OOI_HEAD_CODE_OF_WAY_OF_DELIVERY	48	C	8		RW	Kod för leveranssätt. Leveranssätt definieras i ett eget register.
ADK_OOI_HEAD_TEXT1	49	C	60		RW	Fri text 1.
ADK_OOI_HEAD_TEXT2	50	C	60		RW	Fri text 2.
ADK_OOI_HEAD_TEXT3	51	C	60		RW	Fri text 3.
ADK_OOI_HEAD_SET_OF_VAT1	52	D	8	5	RW	Momssats för momskod 1. Lagras i fakturan för att se den momssats som gällde när fakturan skapades.
ADK_OOI_HEAD_SET_OF_VAT2	53	D	8	5	RW	Som SET_OF_VAT1 men för momskod 2.
ADK_OOI_HEAD_SET_OF_VAT3	54	D	8	5	RW	Som SET_OF_VAT1 men för momskod 3.
ADK_OOI_HEAD_SET_OF_VAT4	55	D	8	5	RW	Som SET_OF_VAT1 men för momskod 4
ADK_OOI_HEAD_CARGO_AMOUNT	56	D	10	2	RW	Belopp för frakt i vald valuta.
ADK_OOI_HEAD_CARGO_VAT_CODE	57	C	1		RW	Momskod för frakt.
ADK_OOI_HEAD_DISPATCH_FEE	58	D	10	2	RW	Expeditionsavgift i vald valuta.
ADK_OOI_HEAD_VAT_CODE_DISPATCH_FEE	59	C	1		RW	Momskod för expeditionsavgift.
ADK_OOI_HEAD_ROUND_OFF	60	D	7	2	R	Öresavrundning i vald valuta.
ADK_OOI_HEAD_TOTAL_AMOUNT	61	D	14	2	R	Totalbelopp inkl eventuell moms. Belopp i vald valuta. Är positiv även för kreditfakturer.
ADK_OOI_HEAD_BALANCE	62	D	14	2	R	Saldo. Kundfordran i vald valuta. Positivt för vanliga fakturer, negativt för kreditfakturer. Används enbart för fakturer.
ADK_OOI_HEAD_PROJECT_CODE	63	C	10		RW	Kod för projekt. Projekt definieras i ett eget register.
ADK_OOI_HEAD_PROFIT_CENTRE	64	C	6		RW	Kod för resultatenhet. Resultatenheter definieras i ett eget register.
ADK_OOI_HEAD_BASIS_OF_VAT0	65	D	14	2	R	Momsunderlag för momskod 0.
ADK_OOI_HEAD_BASIS_OF_VAT1	66	D	14	2	R	Momsunderlag för momskod 1.
ADK_OOI_HEAD_BASIS_OF_VAT2	67	D	14	2	R	Momsunderlag för momskod 2.
ADK_OOI_HEAD_BASIS_OF_VAT3	68	D	14	2	R	Momsunderlag för momskod 3.
ADK_OOI_HEAD_BASIS_OF_VAT4	69	D	14	2	R	Momsunderlag för momskod 4.
ADK_OOI_HEAD_AMOUNT_OF_VAT1	70	D	14	2	R	Momsbelopp för momskod 1.
ADK_OOI_HEAD_AMOUNT_OF_VAT2	71	D	14	2	R	Momsbelopp för momskod 2.
ADK_OOI_HEAD_AMOUNT_OF_VAT3	72	D	14	2	R	Momsbelopp för momskod 3.
ADK_OOI_HEAD_AMOUNT_OF_VAT4	73	D	14	2	R	Momsbelopp för momskod 4.
ADK_OOI_HEAD_PENALTY_INTEREST_RATE	74	D	6	2	RW	Dröjsmålsräntesats (procent).
ADK_OOI_HEAD_CUSTOMER_DEMAND_ACCOUNT	75	C	6		RW	Kontonummer för kundfordran.
ADK_OOI_HEAD_CARGO_ACCOUNT	76	C	6		RW	Kontonummer för frakt.
ADK_OOI_HEAD_DISPATCH_FEE_ACCOUNT	77	C	6		RW	Kontonummer för expeditionsavgift.
ADK_OOI_HEAD_BALANCE2	78	D	14	2	R	Saldo i inhemsk valuta.
ADK_OOI_HEAD_NUMBER_OF_ORIGINAL_INVOICE	80	D	0		R	Den faktura som en kreditfaktura krediterar.
ADK_OOI_HEAD_VAT1_ACCOUNT	81	C	15	0	RW	Konto för momskod 1.
ADK_OOI_HEAD_VAT2_ACCOUNT	82	C	6		RW	Konto för momskod 2.
ADK_OOI_HEAD_VAT3_ACCOUNT	83	C	6		RW	Konto för momskod 3.
ADK_OOI_HEAD_VAT4_ACCOUNT	84	C	6		RW	Konto för momskod 4.
ADK_OOI_HEAD_ROUND_OFF_ACCOUNT	85	C	6		RW	Konto för avrundning.

ADK_OOI_HEAD_DISCOUNT_ACCOUNT	86	C	6		RW	Konto för givna rabatter.
ADK_OOI_HEAD_CUSTOMER_INVOICE_VAT_NUMBER	87	C	6		RW	Momsregistreringsnummer för kunden på fakturan. VAT-nr
ADK_OOI_HEAD_SUMMARY_INVOICE (1)	88	L	1		RW	Order: Samlingsfakturerad ordern.
ADK_OOI_HEAD_EU_ACCOUNT_FOR_QUARTERLY	89	L	1		RW	Sann: Fakturan ska kvartalsredovisas till EU.
ADK_OOI_HEAD_EU_THIRD_PART_TRADE	90	L	1		RW	Sann: Fakturan är ingår i trepartshandel med EU-länder.
ADK_OOI_HEAD_SAVE_TEXT	93	L	1		RW	Sann: Behåll innehållet i textraderna (TEXT1, TEXT2, TEXT3) när order eller faktura skapas från offert respektive order.
ADK_OOI_HEAD_EXTRA_DOCUMENT_PRINTED	94	L	1		RW	Sann: Extra dokument utskrivet.
ADK_OOI_HEAD_BOOK_ORDER (1)	95	L	1		R	Sann: En order har skapats från detta dokument (beställning)
ADK_OOI_HEAD_GROSS	96	D	14	2	R	Brutto.
ADK_OOI_HEAD_NET	97	D	14	2	R	Netto. Brutto efter rabatt.
ADK_OOI_HEAD_EXCLUDING_OF_VAT	98	D	14	2	R	Totalbelopp exklusive eventuell moms i fakturans valuta
ADK_OOI_HEAD_VAT_AMOUNT	99	D	14	2	R	Total momsbelopp. Summa av AMOUNT_OF_VAT1 till 3
ADK_OOI_HEAD_CONTRIBUTION_MARGIN	100	D	14	2	R	Täckningsbidrag efter fakturarabatt. Detta fält används för summering till kundens täckningsbidrag.
ADK_OOI_HEAD_VALUE	101	D	14	2	R	Ordervärdet efter fakturarabatt. Detta fält används för summering till kundens omsättning.
ADK_OOI_HEAD_TYPE_IN_TEXT	102	C	14	2	R	Returnerar långa beteckning på fakturatypen: 'Faktura', 'Kredit', 'Kontantnota', 'Ränta', 'Man faktura' eller 'Man kredit'.
ADK_OOI_HEAD_ROWS	103	Data	14	2	RW	Pekar ut tillhörande rader
ADK_OOI_HEAD_NROWS	104	D	14	2	RW	Antalet tillhörande rader
ADK_OOI_HEAD_LOCAL_REMARK	105	C	12		RW	Kommentar (tidigare benämnt Lokal anmärkning)
ADK_OOI_HEAD_CREDIT_INVOICE_COPY_NUMBER	106	D	1		RW	Faktura att kreditera vid skapande av kreditfaktura.
ADK_OOI_HEAD_CREDIT_INVOICE_COPY_WHAT	107	D	9	0	RW	Vad som ska kopieras vid skapande av kreditfaktura. Se ADK_CREDIT_INVOICE_TO_COPY för specification i adk.h.
ADK_OOI_HEAD_EANLOC	108	C	120		RW	EAN-lokaliseringsnummer. Alternativt EAN-lokaliseringsnummer för leveransadress.
ADK_OOI_HEAD_DELIVERY_EANLOC	109	C	15	0	RW	Alternativt EAN-lokaliseringsnummer för leveransadress.
ADK_OOI_HEAD_DISPATCHER	110	C	1	0	RW	Kod för speditör. Speditör definieras i ett eget register.
ADK_OOI_HEAD_CREDIT_NOTE_STORAGE_AFFECT	111	L	1		R	Sann: Kreditfakturer påverka lagret.
ADK_OOI_HEAD_OCR_NUMBER	112	C	20		R	OCR-nr på faktura
ADK_OOI_HEAD_CODE_OF_COUNTRY	113	C	7		RW	Landskod
ADK_OOI_HEAD_DELIVERY_CODE_OF_COUNTRY	114	C	1		RW	Landskod leveransadress
ADK_OOI_HEAD_INVOICE_BASE	115	D	25		RW	Orderunderlagsnr
ADK_OOI_HEAD_CONTRACTNR	116	C	4		RW	Kontraktsnr
ADK_OOI_HEAD_TOT_SHIPWT	117	D	14	2	RW	Total sändvikt
ADK_OOI_HEAD_DOCUMENT_DATE3	118	Date	15	0	RW	Används inte i order
ADK_OOI_HEAD_BUILDSERVICE	119	L	1		RW	Används för byggtjänst
ADK_OOI_HEAD_LAST_PAYMENT	120	Date	14	2	R	Används inte i order
ADK_OOI_HEAD_CASH_RECEIPT	121	L	1		RW	Används inte i order
ADK_OOI_HEAD_STOCK_SHIP_VER_DATE	122	Date	1		RW	Det datum då leverans ska lagerbokföras
ADK_OOI_HEAD_REduced_GENERAL_PENSION_FEE	123	L	1		RW	Används inte från version 4.5

ADK_OOI_HEAD_CREATED_I NVCOM	124	L	1		R	Dokumentet skapat med Visma Byråstöd aktiverat. Order: Används ej
ADK_OOI_HEAD_CREATED_S END	125	L	1		R	Dokumentet skickat med Visma Byråstöd. Order: Används ej
ADK_OOI_HEAD_CREATED_R ECEIVED	126	L	1		R	Dokumentet hämtat med Visma Byråstöd. Order: Används ej
ADK_OOI_HEAD_TAX_REDUC TION_INCLUDED	127	L	1		RW	Innehåller husarbete
ADK_OOI_HEAD_TAX_REDUC TION_MANUALLY	128	L	1		RW	Manuell fördelning (skattereduktion)
ADK_OOI_HEAD_TAX_REDUC TION_TAX_BASIS	129	D	14	2	R	Undelag för skattereduktion
ADK_OOI_HEAD_TAX_REDUC TION	130	D	14	2	R	Skattereduktion
ADK_OOI_HEAD_TAX_REDUC TION_NUMBER_OF_PERSON S	131	D	1		R	Antal personer (fördelad skattereduktion)
ADK_OOI_HEAD_TAX_REDUC TION_BUILDING_WORK	132	L	1		RW	Ordern avser ROT-arbete
ADK_OOI_HEAD_TAX_REDUC TION_HOUSING_COOPERATI VE_IDENTITY_NUMBER	133	C	14	2	RW	Bostadsrättsföreningens organisationsnummer
ADK_OOI_HEAD_TAX_REDUC TION_DESCRIPTION_OF_PRO PERTY	134	C	4	0	RW	Fastighetsbeteckning
ADK_OOI_HEAD_TAX_REDUC TION_AMOUNT_TO_PAY	135	D	14	2	R	Belopp att betala (efter skattereduktion)
ADK_OOI_HEAD_AGREEMEN T_DATE_TYPE	136	C	14		R	Order: Används ej
ADK_OOI_HEAD_TIMESTAMP	137	Date	100		R	Tidsstämpel
ADK_OOI_HEAD_CUSTOMER _REFERENCE_SIGN	138	C	14	2	RW	Kundrefsignatur
ADK_OOI_HEAD_DELIVERY_I D	139	D	1		RW	Id för avvikande leveransadress
ADK_OOI_HEAD_ORDER_BA SE	140	D	10		R	Order: Orderunderlagsnummer vid import
ADK_OOI_HEAD_ORDER_BA SE_TYPE	141	C	8		R	Order: Orderunderlagstyp vid import
ADK_OOI_HEAD_ELECTRONI C_REFERENCE	142	C	35	0	RW	Referenskod
ADK_OOI_HEAD_PRINTOPTIO N (7)	143	D	15	0	RW	Utskriftsval för dokumentet
ADK_OOI_HEAD_PRINTOPTIO N_DELIVERY_NOTE (7)	144	D	1		RW	Order: Utskriftsval följesedel
ADK_OOI_HEAD_PRINTOPTIO N_EXTRA_DOCUMENT (7)	145	D	35		RW	Utskriftsval för extradokumentet
ADK_OOI_HEAD_ELECTRONI C_INVOICE_ATTACHMENT	146	C	160	0	R	Order: används ej
ADK_OOI_HEAD_AI_IMPORTE D_XML_NAME	147	C	4	0	R	Order: Filnamn för xmlfil när e-ordern är importerad. Filen finns i katalogen SPCSXML_IMPARKIV i företagskatalogen.
ADK_OOI_HEAD_TAX_REDUC TION_PROC (3)	148	D	4	0	RW	Procentsats för RUT/ROT
ADK_OOI_HEAD_TAX_REDUC TION_MAXAMOUNT (3)	149	D	9	0	RW	Maxbelopp för RUT/ROT per person
ADK_OOI_HEAD_TAX_REDUC TION_GREEN_TECHNOLOGY (3)	150	L	1		RW	Innehåller skattereduktion för grön teknik.
ADK_OOI_HEAD_TAX_REDUC TION_ALL_TYPES (3)	151	L	1		RW	Innehåller skattereduktion.

(1) Fältet används endast i Visma Administration 2000.

(6) Fältet är eReadWrite i Visma Administration 2000 och eReadOnly i 500/1000 och Förening

(7) För tillåtna värden läs avsnitt Utskriftsval i kunder, leverantörer och tillhörande dokument ovan

Orderrader – ADK_DB_ORDER_ROW

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
----------	----	-----	-------	-----	------	-------------

ADK_OOI_ROW_ARTICLE_NUMBER	0	C	40		RW	Artikelnummer
ADK_OOI_ROW_SUPPLIER_ARTICLE_NUMBER	1	C	40		R	Leverantörens artikelnummer. Används ej i order.
ADK_OOI_ROW_QUANTITY1	2	D	12	4	RW	Antalsfält. Order: Levererat antal.
ADK_OOI_ROW_QUANTITY2	3	D	12	4	RW	Extra antalsfält. Order: Beställt antal.
ADK_OOI_ROW_QUANTITY3	4	D	12	4	RW	Extra antalsfält. Order: Restnoterat antal.
ADK_OOI_ROW_PRICE_EACH_CURRENT_CURRENCY	5	D	12	4/2	RW	Pris per enhet i aktuell valuta. Inklusive eller exklusive moms.
ADK_OOI_ROW_PRICE2	6	D	12	4/2	RW	Extra fält för pris. I offert, order, och faktura används detta fält för kalkylpris per enhet utan moms. Används för att visa beräknat täckningsbidrag.
ADK_OOI_ROW_DISCOUNT	7	D	10	2	RW	Rabatt i procent eller belopp. Flagga indikerar vilket. Alltid positivt.
ADK_OOI_ROW_AMOUNT_DISCOUNT_FLAG	8	L	1		RW	Sann indikerar beloppsrabatt, annars procentrabatt. Alltid positivt.
ADK_OOI_ROW_AMOUNT_CURRENT_CURRENCY	9	D	14	2	RW	Totalt belopp i aktuell valuta inklusive eller exklusive moms. Normalt positivt.
ADK_OOI_ROW_AMOUNT_DOMESTIC_CURRENCY	10	D	14	2	R	Totalt belopp i inhemsk valuta utan moms. Normalt positivt. Order: Radens ordervärde i inhemsk valuta utan moms.
ADK_OOI_ROW_ACCOUNT_NUMBER	11	C	6		RW	Kontonummer.
ADK_OOI_ROW_PROFIT_CENTER	12	C	6		RW	Resultatenhet. Definieras i eget register.
ADK_OOI_ROW_PROJECT	13	C	10		RW	Projekt. Definieras i eget register.
ADK_OOI_ROW_TEXT	14	C	60		RW	Används för text. Exempelvis artikelbenämning i offert, order, faktura.
ADK_OOI_ROW_UNIT	15	C	4		RW	Enhetskod. Definieras i eget register.
ADK_OOI_ROW_VAT_CODE	16	C	1		RW	Momskod. Påverkas inte om inkl moms eller ej.
ADK_OOI_ROW_NOTE_OF_OUTSTANDING_ORDERS (1)	17	L	1		RW	Används i order för att indikera att differens mellan beställt och levererat inte ska restnoteras.
ADK_OOI_ROW_DATE2	18	Date	8		RW	Extra datumfält. Används i order för beräknat leveransdatum och i faktura för restnoterat leveransdatum.
ADK_OOI_ROW_TYPE_OF_ROW	19	C	1		RW	Radtyp. Användning beror på typ, normalt blank. 'T' = Extrad, 'D' – deltotal (på samlingsfaktura rad med ordersumma) 'R' – Rånterad.
ADK_OOI_ROW_CONTRIBUTION_DEGREE	20	D	6	1	R	Täckningsgrad.
ADK_OOI_ROW_CONTRIBUTION_MARGIN	21	D	14	2	R	Täckningsbidrag. Blankas om noll.
ADK_OOI_ROW_PRINT (1)	22	L	1		RW	Anger om raden ska skrivas ut (gäller offert, order eller faktura).
ADK_OOI_ROW_PRINT2 (1)	23	L	1		RW	Anger om raden ska skrivas ut (gäller följesedel i en order).
	24	U	0		R	Används inte
ADK_OOI_ROW_BAR_CODE	25	C	32		RW	Streckkod.
ADK_OOI_ROW_DATE3	26	Date	8		R	Används inte i order
ADK_OOI_ROW_ARTICLE_TYPE	27	C	1		R	Används inte i offert
ADK_OOI_ROW_FROM_TYPE	28	C	1		R	Används inte i order
ADK_OOI_ROW_FROM_DOCUMENT	29	D	15	0	R	Används inte i order
ADK_OOI_ROW_FROM_DOCUMENT	30	D	9	0	R	Används inte i order
ADK_OOI_ROW_ROWNUMBER	31	D	9	0	R	Radnummer
ADK_OOI_ROW_CONNECTION_TYPE	32	C	3		R	Kopplingstyp
ADK_OOI_ROW_CONNECTION_DOCUMENT	33	D	15	0	R	Kopplingsdokument
ADK_OOI_ROW_CONNECTION_DOCUMENT	34	D	9	0	R	Kopplingsnummer
ADK_OOI_ROW_SERIAL_NUMBER	35	C	30		RW	Serienummer

ADK_OOI_ROW_BATCH_NUMBER	36	C	30		RW	Batchnummer
ADK_OOI_ROW_BEST_BEFORE	37	Date	8		RW	Bäst före datum
ADK_OOI_ROW_NOTIFY_QT1	38	D	12	4	R	Används inte i order
ADK_OOI_ROW_NOTIFY_QT2	39	D	12	4	R	Används inte i order
ADK_OOI_ROW_PIN_BOOKING (1)	40	L	1		R	Häftad textrad
ADK_OOI_ROW_NUM_PIN_BOOKING (1)	41	D	9	0	R	Antalet häftade textrader
ADK_OOI_ROW_LABOUR_COST	42	L	1		RW	Arbetskostnad
ADK_OOI_ROW_QT_BOOKING	43	D	12	3	R	Kvantitet (används av leverantörsfaktura)
ADK_OOI_ROW_ORDER_BASE_TYPE	44	C	1		R	Order: Orderunderlagstyp
ADK_OOI_ROW_USER_ID	45	C	15		R	Medarbetarid
ADK_OOI_ROW_STRUCTURE_ORDER (1)	46	D	9	0	R	Ordning i struktur
ADK_OOI_ROW_PACKAGE (1)	47	L	1		R	Paket
ADK_OOI_ROW_STRUCTURE (1)	48	L	1		R	Struktur
ADK_OOI_ROW_TIMESTAMP	49	Date	10		R	Tidsstämpel
ADK_OOI_ROW_TAX_REDUCTION_TYPES_ID	50	C	4		RW	Husarbetstypid.
ADK_OOI_ROW_HOURS_WORKED	51	D	7	1	R	Används inte i order.
ADK_OOI_ROW_H_SCHAB_PROC	52	D	6	2	RW	Schablonprocentsats för vissa husarbetstyper och grön teknik.

(3) Fältet används endast i Visma Administration 2000.

Fakturor – ADK_DB_INVOICE_HEAD

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_OOI_HEAD_DOCUMENT_NUMBER	0	D	15	0	RW	Dokumentnummer från löpnummerserie. Unikt. Anger offertnummer, ordernummer eller fakturanummer.
ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER1	1	D	15	0	RW/R (6)	Vidarekoppling till dokumentnummer i kedjan Offert – Order – Faktura. Faktura: Ordernummer. 0 om direktfaktura eller samlingsfaktura.
ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER2	2	D	15	0	R	Vidarekoppling till dokumentnummer i kedjan Offert – Order – Faktura. Faktura: Avtalsnummer, fungerar också som identifikation på att det är en faktura från ett avtal.
ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER3	3	D	15	0	R	Faktura: Journalnummer.
ADK_OOI_HEAD_DOCUMENT_CONNECTION_NUMBER4	4	D	15	0	R	Faktura: Verifikationsnummer.
ADK_OOI_HEAD_DOCUMENT_NOT_DONE	5	L	1		RW	Sann indikerar att dokumentet inte är slutbehandlat (ej utskriftsklart).
ADK_OOI_HEAD_DOCUMENT_PRINTED	6	L	1		RW	Faktura: Faktura utskriven.
ADK_OOI_HEAD_DOCUMENT_CANCELLED	7	L	1		RW	Sann anger att dokumentet är makulerat.
ADK_OOI_HEAD_DOCUMENT_DATE1	8	Date	8		RW	Faktura: Fakturadatum.
ADK_OOI_HEAD_DOCUMENT_DATE2	9	Date	8		RW	Faktura: Förfallodatum.
ADK_OOI_HEAD_ORDER_DELIVERED	10	L	1		R	Sann anger att leverans har skett på ordern, det innebär att artiklar anses utplockade från lager.
ADK_OOI_HEAD_ORDER_PICK_PRINTED	11	L	1		R	Sann anger att plocklista är utskriven från ordern.
ADK_OOI_HEAD_ORDER_DELIVERY_NOTE_PRINTED	12	L	1		R	Sann anger att följesedel är utskriven från ordern.
ADK_OOI_HEAD_TYPE_OF_INVOICE	13	C	1		RW	Fakturatyp. F = normal faktura, K = kreditfaktura, N = kontantnota, R = räntefaktura, f = manuell faktura, k = manuell kreditfaktura. Endast möjligt

						att skapa normalfakturer, kontantnotor och kreditfakturer.
ADK_OOI_HEAD_CUSTOMER_NUMBER	14	C	16		RW	Kundnummer.
ADK_OOI_HEAD_CUSTOMER_NAME	15	C	50		RW	Kundnamn.
ADK_OOI_HEAD_MAILING_ADDRESS1	16	C	35		RW	Postadress.
ADK_OOI_HEAD_MAILING_ADDRESS2	17	C	35		RW	Postadress, rad 2.
ADK_OOI_HEAD_ZIPCODE	18	C	12		RW	Postnr.
ADK_OOI_HEAD_CITY	19	C	24		RW	Ortsnamn.
ADK_OOI_HEAD_COUNTRY	20	C	24		RW	Land.
ADK_OOI_HEAD_DELIVERY_NAME	21	C	50		RW	Namn för leverans. Om blankt ska innehållet i fältet NAME användas.
ADK_OOI_HEAD_DELIVERY_MAILING_ADDRESS1	22	C	35		RW	Postadress, gata, box för leverans. Om blankt ska innehållet i fältet MAILING_ADDRESS1 användas.
ADK_OOI_HEAD_DELIVERY_MAILING_ADDRESS2	23	C	35		RW	Postadress, gata, box för leverans, rad 2. Om blankt ska innehållet i fältet MAILING_ADDRESS2 användas.
ADK_OOI_HEAD_DELIVERY_ZIPCODE	24	C	12		RW	Postnr. Om blankt ska innehållet i fältet ZIPCODE användas.
ADK_OOI_HEAD_DELIVERY_CITY	25	C	24		RW	Ortsnamn för leverans. Om blankt ska innehållet i fältet CITY användas.
ADK_OOI_HEAD_DELIVERY_COUNTRY	26	C	24		RW	Land för leverans. Om blankt ska innehållet i fältet COUNTRY användas.
ADK_OOI_HEAD_CUSTOMER_REFERENCE_NAME	27	C	50		RW	Referensperson hos kund.
ADK_OOI_HEAD_CUSTOMER_ORDER_NUMBER	28	C	25		RW	Order, faktura: Kundens ordernr.
ADK_OOI_HEAD_CUSTOMER_REFERENCE_NUMBER (1)	29	C	16		R	Används vid offert.
ADK_OOI_HEAD_OUR_REFERENCE_NAME	30	C	25		RW	Vår referensperson. Om företags-/föreningsinställning för vår ref inte är blank laddas denna hit, annars laddas namn på säljare.
ADK_OOI_HEAD_DISTRICT_CODE (2)	31	C	8		RW	Kod för distrikt. Distrikt definieras i ett eget register.
ADK_OOI_HEAD_SELLER_CODE (2)	32	C	10		RW	Kod för säljare. Säljare definieras i ett eget register.
ADK_OOI_HEAD_CURRENCY_CODE (3)	33	C	4		RW	Valutakod. Valutor definieras i ett eget register.
ADK_OOI_HEAD_LANGUAGE_CODE (3)	34	C	4		RW	Språkkod. Språk definieras i ett eget register.
ADK_OOI_HEAD_CURRENCY_RATE (3)	35	D	12	6	RW	Valutakurs. Om valutakursen är noll ska den betraktas som ett.
ADK_OOI_HEAD_CURRENCY_UNIT (3)	36	D	9	0	R	Valutaenhet. Om valutaenheten är noll ska den betraktas som ett.
ADK_OOI_HEAD_REMINDER	37	L	1		RW	Sann anger att kravbrev ska kunna sändas.
ADK_OOI_HEAD_DEMAND_FEE	38	L	1		RW	Sann anger att kravavgift ska kunna tas ut av kunden.
ADK_OOI_HEAD_INVOICE_INTEREST	39	L	1		RW	Sann anger att räntefakturering ska kunna göras.
ADK_OOI_HEAD_EXPORT	40	L	1		RW	Sann anger export.
ADK_OOI_HEAD_INCLUDING_VAT	41	L	1		RW	Sann anger att alla priser är inkl moms.
ADK_OOI_HEAD_DO_NOT_CREATE_BACKORDER (1)	42	L	1		RW	Sann anger att restorder ej ska kunna skapas om artiklar saknas.
ADK_OOI_HEAD_LIST_OF_PRICE_CODE	43	C	8		RW	Kod för prislista. Finns rabattavtal har detta fält ingen funktion. Prislistor definieras i ett eget register.
ADK_OOI_HEAD_INVOICE_DISCOUNT	44	D	5	1	RW	Fakturarakbatt i procent. Finns rabattavtal har detta fält ingen funktion.
ADK_OOI_HEAD_CODE_OF_DISCOUNT	45	C	16		RW	Kod för rabattavtal. Rabattavtal definieras i ett eget register.
ADK_OOI_HEAD_CODE_OF_TERMS_OF_PAYMENT	46	C	8		RW	Kod för betalvillkor. Betalvillkor definieras i ett eget register.
ADK_OOI_HEAD_CODE_OF_TERMS_OF_DELIVERY	47	C	8		RW	Kod för leveransvillkor. Leveransvillkor definieras i ett eget register.
ADK_OOI_HEAD_CODE_OF_WAY_OF_DELIVERY	48	C	8		RW	Kod för leveranssätt. Leveranssätt definieras i ett eget register.
ADK_OOI_HEAD_TEXT1	49	C	60		RW	Fri text 1.
ADK_OOI_HEAD_TEXT2	50	C	60		RW	Fri text 2.

ADK_OOI_HEAD_TEXT3	51	C	60		RW	Fri text 3.
ADK_OOI_HEAD_SET_OF_VAT1	52	D	8	5	RW	Momssats för momskod 1. Lagras i fakturan för att se den momssats som gällde när fakturan skapades.
ADK_OOI_HEAD_SET_OF_VAT2	53	D	8	5	RW	Som SET_OF_VAT1men för momskod 2.
ADK_OOI_HEAD_SET_OF_VAT3	54	D	8	5	RW	Som SET_OF_VAT1men för momskod 3.
ADK_OOI_HEAD_SET_OF_VAT4	55	D	8	5	RW	Som SET_OF_VAT1men för momskod 4
ADK_OOI_HEAD_CARGO_AMOUNT	56	D	10	2	RW	Belopp för frakt i vald valuta.
ADK_OOI_HEAD_CARGO_VAT_CODE	57	C	1		RW	Momskod för frakt.
ADK_OOI_HEAD_DISPATCH_FEE	58	D	10	2	RW	Expeditionsavgift i vald valuta.
ADK_OOI_HEAD_VAT_CODE_DISPATCH_FEE	59	C	1		RW	Momskod för expeditionsavgift.
ADK_OOI_HEAD_ROUND_OFF	60	D	7	2	R	Öresavrundning i vald valuta.
ADK_OOI_HEAD_TOTAL_AMOUNT	61	D	14	2	R	Totalbelopp inkl eventuell moms. Belopp i vald valuta. Är positiv även för kreditfakturer.
ADK_OOI_HEAD_BALANCE	62	D	14	2	R	Saldo. Kundfordran i vald valuta. Positivt för vanliga fakturer, negativt för kreditfakturer. Används enbart för fakturer.
ADK_OOI_HEAD_PROJECT_CODE (4)	63	C	10		RW	Kod för projekt. Projekt definieras i ett eget register.
ADK_OOI_HEAD_PROFIT_CENTRE	64	C	6		RW	Kod för resultatenhet. Resultatenheter definieras i ett eget register.
ADK_OOI_HEAD_BASIS_OF_VAT0	65	D	14	2	R	Momsunderlag för momskod 0.
ADK_OOI_HEAD_BASIS_OF_VAT1	66	D	14	2	R	Momsunderlag för momskod 1.
ADK_OOI_HEAD_BASIS_OF_VAT2	67	D	14	2	R	Momsunderlag för momskod 2.
ADK_OOI_HEAD_BASIS_OF_VAT3	68	D	14	2	R	Momsunderlag för momskod 3.
ADK_OOI_HEAD_BASIS_OF_VAT4	69	D	14	2	R	Momsunderlag för momskod 4.
ADK_OOI_HEAD_AMOUNT_OF_VAT1	70	D	14	2	R	Momsbelopp för momskod 1.
ADK_OOI_HEAD_AMOUNT_OF_VAT2	71	D	14	2	R	Momsbelopp för momskod 2.
ADK_OOI_HEAD_AMOUNT_OF_VAT3	72	D	14	2	R	Momsbelopp för momskod 3.
ADK_OOI_HEAD_AMOUNT_OF_VAT4	73	D	14	2	R	Momsbelopp för momskod 4.
ADK_OOI_HEAD_PENALTY_INTEREST_RATE	74	D	6	2	RW	Dröjsmålsräntesats (procent).
ADK_OOI_HEAD_CUSTOMER_DEMAND_ACCOUNT	75	C	6		RW	Kontonummer för kundfordran.
ADK_OOI_HEAD_CARGO_ACCOUNT	76	C	6		RW	Kontonummer för frakt.
ADK_OOI_HEAD_DISPATCH_FEE_ACCOUNT	77	C	6		RW	Kontonummer för expeditionsavgift.
ADK_OOI_HEAD_BALANCE2	78	D	14	2	R	Saldo i inhemsk valuta.
ADK_OOI_HEAD_NUMBER_OF_ORIGINAL_INVOICE	80	D	0		R	Den faktura som en kreditfaktura krediterar.
ADK_OOI_HEAD_VAT1_ACCOUNT	81	C	15	0	RW	Konto för momskod 1.
ADK_OOI_HEAD_VAT2_ACCOUNT	82	C	6		RW	Konto för momskod 2.
ADK_OOI_HEAD_VAT3_ACCOUNT	83	C	6		RW	Konto för momskod 3.
ADK_OOI_HEAD_VAT4_ACCOUNT	84	C	6		RW	Konto för momskod 4.
ADK_OOI_HEAD_ROUND_OFF_ACCOUNT	85	C	6		RW	Konto för avrundning.
ADK_OOI_HEAD_DISCOUNT_ACCOUNT	86	C	6		RW	Konto för givna rabatter.
ADK_OOI_HEAD_CUSTOMER_INVOICE_VAT_NUMBER	87	C	6		RW	Momsregistreringsnummer för kunden på fakturan. VAT-nr

ADK_OOI_HEAD_SUMMARY_INVOICE (1)	88	L	1		R	Faktura: Fakturan är en samlingsfaktura.
ADK_OOI_HEAD_EU_ACCOUNT_FOR_QUARTERLY	89	L	1		RW	Sann: Fakturan ska kvartalsredovisas till EU.
ADK_OOI_HEAD_EU_THIRD_PART_TRADE	90	L	1		RW	Sann: Fakturan är ingår i trepartshandel med EU-länder.
ADK_OOI_HEAD_SAVE_TEXT	93	L	1		R	Sann: Behåll innehållet i textraderna (TEXT1, TEXT2, TEXT3) när order eller faktura skapas från offert respektive order.
ADK_OOI_HEAD_EXTRA_DOCUMENT_PRINTED	94	L	1		RW	Sann: Extra dokument utskrivet.
ADK_OOI_HEAD_BOOK_ORDER (1)	95	L	1		R	Sann: En order har skapats från detta dokument (beställning)
ADK_OOI_HEAD_GROSS	96	D	14	2	R	Brutto.
ADK_OOI_HEAD_NET	97	D	14	2	R	Netto. Brutto efter rabatt.
ADK_OOI_HEAD_EXCLUDING_OF_VAT	98	D	14	2	R	Totalbelopp exklusive eventuell moms i fakturans valuta
ADK_OOI_HEAD_VAT_AMOUNT	99	D	14	2	R	Total momsbelopp. Summa av AMOUNT_OF_VAT1 till 3
ADK_OOI_HEAD_CONTRIBUTION_MARGIN	100	D	14	2	R	Täckningsbidrag efter fakturarabatt. Detta fält används för summering till kundens täckningsbidrag.
ADK_OOI_HEAD_VALUE	101	D	14	2	R	Ordervärdet efter fakturarabatt. Detta fält används för summering till kundens omsättning.
ADK_OOI_HEAD_TYPE_IN_TEXT	102	C	14	2	R	Returnerar långa beteckning på fakturatypen: 'Faktura', 'Kredit', 'Kontantnota', 'Ränta', 'Man faktura' eller 'Man kredit'.
ADK_OOI_HEAD_ROWS	103	Data	14	2	RW	Pekar ut tillhörande rader
ADK_OOI_HEAD_NROWS	104	D	14	2	RW	Antalet tillhörande rader
ADK_OOI_HEAD_LOCAL_REMARK	105	C	12		RW	Kommentar (tidigare benämnt Lokal anmärkning)
ADK_OOI_HEAD_CREDIT_INVOICE_COPY_NUMBER	106	D	1		RW	Faktura att kreditera vid skapande av kreditfaktura.
ADK_OOI_HEAD_CREDIT_INVOICE_COPY_WHAT	107	D	9	0	RW	Vad som ska kopieras vid skapande av kreditfaktura. Se ADK_CREDIT_INVOICE_TO_COPY för specification i adk.h.
ADK_OOI_HEAD_EANLOC (3)	108	C	120		RW	EAN-lokaliseringsnummer. Alternativt EAN-lokaliseringsnummer för leveransadress.
ADK_OOI_HEAD_DELIVERY_EANLOC (3)	109	C	15	0	RW	Alternativt EAN-lokaliseringsnummer för leveransadress.
ADK_OOI_HEAD_DISPATCHER (3)	110	C	1	0	RW	Kod för speditör. Speditör definieras i ett eget register.
ADK_OOI_HEAD_CREDIT_NOTE_STORAGE_AFFECT	111	L	1		RW	Sann: Kreditfakturor påverka lagret.
ADK_OOI_HEAD_OCR_NUMBER	112	C	20		R	OCR-nr på faktura
ADK_OOI_HEAD_CODE_OF_COUNTRY (3)	113	C	7		RW	Landskod
ADK_OOI_HEAD_DELIVERY_CODE_OF_COUNTRY (3)	114	C	1		RW	Landskod leveransadress
ADK_OOI_HEAD_INVOICE_BASE (3)	115	D	25		RW	Fakturaunderlagsnr
ADK_OOI_HEAD_CONTRACTNR	116	C	4		RW	Kontraktsnr
ADK_OOI_HEAD_TOT_SHIPWT	117	D	14	2	RW	Används inte i faktura
ADK_OOI_HEAD_DOCUMENT_DATE3	118	Date	15	0	RW	Leveransdatum
ADK_OOI_HEAD_BUILDSERVICE	119	L	1		RW	Används för byggtjänst
ADK_OOI_HEAD_LAST_PAYMENT	120	Date	14	2	R	Senaste betaldatum
ADK_OOI_HEAD_CASH_RECEIPT	121	L	1		RW	Kassakvitto
ADK_OOI_HEAD_STOCK_SHIP_VER_DATE	122	Date	1		RW	Används inte på faktura
ADK_OOI_HEAD_REduced_GENERAL_PENSION_FEE	123	L	1		RW	Används inte från version 4.5
ADK_OOI_HEAD_CREATED_INVCOM	124	L	1		R	Dokumentet skapat med Visma Byråstöd aktiverat.
ADK_OOI_HEAD_CREATED_SEND	125	L	1		R	Dokumentet skickat med Visma Byråstöd.

ADK_OOI_HEAD_CREATED_RECEIVED	126	L	1		R	Dokumentet hämtat med Visma Byråstöd.
ADK_OOI_HEAD_TAX_REDUCTION_INCLUDED	127	L	1		RW	Innehåller husarbete
ADK_OOI_HEAD_TAX_REDUCTION_MANUALLY	128	L	1		RW	Manuell fördelning (skattereduktion)
ADK_OOI_HEAD_TAX_REDUCTION_TAX_BASIS	129	D	14	2	R	Undelag för skattereduktion
ADK_OOI_HEAD_TAX_REDUCTION	130	D	14	2	R	Skattereduktion
ADK_OOI_HEAD_TAX_REDUCTION_NUMBER_OF_PERSONS	131	D	1		R	Antal personer (fördelad skattereduktion)
ADK_OOI_HEAD_TAX_REDUCTION_BUILDING_WORK	132	L	1		RW	Fakturan avser ROT-arbete
ADK_OOI_HEAD_TAX_REDUCTION_HOUSING_COOPERATIVE_IDENTITY_NUMBER	133	C	14	2	RW	Bostadsrättsföreningens organisationsnummer
ADK_OOI_HEAD_TAX_REDUCTION_DESCRIPTION_OF_PROPERTY	134	C	4	0	RW	Fastighetsbeteckning
ADK_OOI_HEAD_TAX_REDUCTION_AMOUNT_TO_PAY	135	D	14	2	R	Belopp att betala (efter skattereduktion)
ADK_OOI_HEAD_AGREEMENT_DATE_TYPE (4)	136	C	14		RW	Typ för datumhantering när man skapar faktura från avtal
ADK_OOI_HEAD_TIMESTAMP	137	Date	100		R	Tidsstämpel
ADK_OOI_HEAD_CUSTOMER_REFERENCE_SIGN	138	C	14	2	RW	Kundrefsignatur
ADK_OOI_HEAD_DELIVERY_ID	139	D	1		RW	Id för avvikande leveransadress
ADK_OOI_HEAD_ORDER_BASE	140	D	10		R	Faktura: Orderunderlagsnummer vid import
ADK_OOI_HEAD_ORDER_BASE_TYPE	141	C	8		R	Faktura: Orderunderlagstyp vid import
ADK_OOI_HEAD_ELECTRONIC_REFERENCE	142	C	35	0	RW	Referenskod
ADK_OOI_HEAD_PRINTOPTION (7)	143	D	15	0	RW	Utskriftsval för dokumentet
ADK_OOI_HEAD_PRINTOPTION_DELIVERY_NOTE (7)	144	D	1		R	Faktura: Används ej
ADK_OOI_HEAD_PRINTOPTION_EXTRA_DOCUMENT (7)	145	D	35		RW	Utskriftsval för extradokumentet
ADK_OOI_HEAD_ELECTRONIC_INVOICE_ATTACHMENT	146	C	160	0	RW	Elektronisk bilaga
ADK_OOI_HEAD_AI_IMPORTERD_XML_NAME	147	C	4	0	R	Faktura: Kan finnas om fakturan är skapad från en order och en e-order som är importerad. Filnamn för xmlfil när e-ordern är importerad. Filen finns i katalogen SPCSXML_IMPARKIV i företagskatalogen.
ADK_OOI_HEAD_TAX_REDUCTION_PROC (3)	148	D	4	0	RW	Procentsats för RUT/ROT
ADK_OOI_HEAD_TAX_REDUCTION_MAXAMOUNT (3)	149	D	9	0	RW	Maxbelopp för RUT/ROT per person
ADK_OOI_HEAD_TAX_REDUCTION_GREEN_TECHNOLOGY (3)	150	L	1		RW	Innehåller skattereduktion för grön teknik.
ADK_OOI_HEAD_TAX_REDUCTION_ALL_TYPES (3)	151	L	1		RW	Innehåller skattereduktion.

(1) Fältet används endast i Visma Administration 2000.

(2) Fältet används endast i Visma Administration 1000/2000.

(3) Fältet används endast i Visma Administration 500/1000/2000.

(4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

(6) Fältet är eReadWrite i Visma Administration 1000/2000 och eReadOnly i 500 och Förening

(7) För tillåtna värden läs avsnitt Utskriftsval i kunder, leverantörer och tillhörande dokument ovan

Fakturarader – ADK_DB_INVOICE_ROW

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
----------	----	-----	-------	-----	------	-------------

ADK_OOI_ROW_ARTICLE_NUMBER	0	C	40		RW	Artikelnummer
ADK_OOI_ROW_SUPPLIER_ARTICLE_NUMBER	1	C	40		R	Leverantörens artikelnummer. Används ej i fakturor.
ADK_OOI_ROW_QUANTITY1	2	D	12	4	RW	Antalsfält. Faktura: Levererat antal.
ADK_OOI_ROW_QUANTITY2 (2)	3	D	12	4	RW	Extra antalsfält. Faktura: Beställt antal.
ADK_OOI_ROW_QUANTITY3 (1)	4	D	12	4	RW	Extra antalsfält. Faktura: Restnoterat antal.
ADK_OOI_ROW_PRICE_EACH_CURRENT_CURRENCY	5	D	12	4/2	RW	Pris per enhet i aktuell valuta. Inklusive eller exklusive moms.
ADK_OOI_ROW_PRICE2	6	D	12	4/2	RW	Extra fält för pris. I offert, order, och faktura används detta fält för kalkylpris per enhet utan moms. Används för att visa beräknat täckningsbidrag.
ADK_OOI_ROW_DISCOUNT	7	D	10	2	RW	Rabatt i procent eller belopp. Flagga indikerar vilket. Alltid positivt.
ADK_OOI_ROW_AMOUNT_DISCOUNT_FLAG	8	L	1		RW	Sann indikerar beloppsrabatt, annars procentrabatt. Alltid positivt.
ADK_OOI_ROW_AMOUNT_CURRENT_CURRENCY	9	D	14	2	RW	Totalt belopp i aktuell valuta inklusive eller exklusive moms. Normalt positivt.
ADK_OOI_ROW_AMOUNT_DOMESTIC_CURRENCY	10	D	14	2	R	Totalt belopp i inhemsk valuta utan moms. Normalt positivt.
ADK_OOI_ROW_ACCOUNT_NUMBER	11	C	6		RW	Kontonummer.
ADK_OOI_ROW_PROFIT_CENTRE	12	C	6		RW	Resultatenhet. Definieras i eget register.
ADK_OOI_ROW_PROJECT (4)	13	C	10		RW	Projekt. Definieras i eget register.
ADK_OOI_ROW_TEXT	14	C	60		RW	Används för text. Exempelvis artikelbenämning i offert, order, faktura.
ADK_OOI_ROW_UNIT	15	C	4		RW	Enhetskod. Definieras i eget register.
ADK_OOI_ROW_VAT_CODE	16	C	1		RW	Momskod. Påverkas inte om inkl moms eller ej.
ADK_OOI_ROW_NOTE_OF_OUTSTANDING_ORDERS (1)	17	L	1		R	Används i order för att indikera att differens mellan beställt och levererat inte ska restnoteras.
ADK_OOI_ROW_DATE2 (6)	18	Date	8		RW	Extra datumfält. Används i faktura för restnoterat leveransdatum. Används i Visma Förening för perioddatum avseende medlemsavgifter.
ADK_OOI_ROW_TYPE_OF_ROW	19	C	1		RW	Radtyp. Användning beror på typ, normalt blank. 'T' = Textrad, 'D' – deltotal (på samlingsfaktura rad med ordersumma) 'R' – Råntraderad.
ADK_OOI_ROW_CONTRIBUTION_DEGREE	20	D	6	1	R	Täckningsgrad.
ADK_OOI_ROW_CONTRIBUTION_MARGIN	21	D	14	2	R	Täckningsbidrag. Blankas om noll.
ADK_OOI_ROW_PRINT (1)	22	L	1		RW	Anger om raden ska skrivas ut (gäller offert, order eller faktura).
ADK_OOI_ROW_PRINT2 (1)	23	L	1		R	Anger om raden ska skrivas ut (gäller följesedel i en order).
	24	U	0		R	Används inte
ADK_OOI_ROW_BAR_CODE (3)	25	C	32		RW	Streckkod.
ADK_OOI_ROW_DATE3	26	Date	8		RW	Fält för leveransdatum
ADK_OOI_ROW_ARTICLE_TYPE (5)	27	C	1		R	Artikeltyp
ADK_OOI_ROW_FROM_TYPE	28	C	1		R	Används inte i faktura
ADK_OOI_ROW_FROM_DOCUMENT	29	D	15	0	R	Används inte i faktura
ADK_OOI_ROW_FROM_DOCUMENT	30	D	9	0	R	Används inte i faktura
ADK_OOI_ROW_ROWNUMBER	31	D	9	0	R	Radnummer
ADK_OOI_ROW_CONNECTION_TYPE	32	C	3		R	Kopplingstyp
ADK_OOI_ROW_CONNECTION_DOCUMENT	33	D	15	0	R	Kopplingsdokument
ADK_OOI_ROW_CONNECTION_DOCUMENT	34	D	9	0	R	Kopplingsnummer
ADK_OOI_ROW_SERIAL_NUMBER	35	C	30		RW	Serienummer

ADK_OOI_ROW_BATCH_NUM BER	36	C	30		RW	Batchnummer
ADK_OOI_ROW_BEST_BEFOR E	37	Date	8		RW	Bäst fore datum
ADK_OOI_ROW_NOTIFY_QT1	38	D	12	4	R	Används inte i faktura
ADK_OOI_ROW_NOTIFY_QT2	39	D	12	4	R	Används inte i faktura
ADK_OOI_ROW_PIN_BOOKIN G	40	L	1		R	Används inte i faktura
ADK_OOI_ROW_NUM_PIN_BO OKING	41	D	9	0	R	Används inte i faktura
ADK_OOI_ROW_LABOUR_CO ST	42	L	1		RW	Arbetskostnad
ADK_OOI_ROW_QT_BOOKING	43	D	12	3	R	Kvantitet (används av leverantörsfaktura)
ADK_OOI_ROW_ORDER_BAS E_TYPE	44	C	1		R	Faktura: Fakturaunderlagstyp
ADK_OOI_ROW_USER_ID	45	C	15		R	Medarbetarid
ADK_OOI_ROW_STRUCTURE _ORDER (1)	46	D	9	0	R	Ordning i struktur
ADK_OOI_ROW_PACKAGE (1)	47	L	1		R	Paket
ADK_OOI_ROW_STRUCTURE (1)	48	L	1		R	Struktur
ADK_OOI_ROW_TIMESTAMP	49	Date	10		R	Tidsstämpel
ADK_OOI_ROW_TAX_REDU CTION_TYPES_ID (3)	50	C	4		RW	Husarbetstypid.
ADK_OOI_ROW_HOURS_WOR KED (3)	51	D	7	1	RW	Arbetade timmar (husarbete).
ADK_OOI_ROW_H_SCHAB_P ROC (3)	52	D	6	2	RW	Schablonprocentsats för vissa husarbetstyper och grön teknik.

(1) Fältet används endast i Visma Administration 2000.

(2) Fältet används endast i Visma Administration 1000/2000.

(3) Fältet används endast i Visma Administration 500/1000/2000.

(4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

(5) Fältet används endast i Visma Förening.

(6) Fältet används endast i Visma Administration 2000 och i Visma Förening.

Avtal – ADK_DB_AGREEMENT_HEAD

Program: Visma Administration 1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_AGREEMENT_HEAD_DO CUMENT_NUMBER	0	D	15	0	RW	Dokumentnummer
ADK_AGREEMENT_HEAD_DO CUMENT_NOT_DONE	1	L	1		RW	Ej Klar
ADK_AGREEMENT_HEAD_DO CUMENT_CANCELLED	2	L	1		RW	Makulerad
ADK_AGREEMENT_HEAD_DO CUMENT_DATE1	3	Date	8		RW	Datum för avtalsregistrering
ADK_AGREEMENT_HEAD_CU STOMER_ORDER_NUMBER	4	C	25		RW	Ert (kundens) ordernummer
ADK_AGREEMENT_HEAD_CU STOMER_NUMBER	5	C	16		RW	Kundnummer
ADK_AGREEMENT_HEAD_CU STOMER_NAME	6	C	50		RW	Kundens namn
ADK_AGREEMENT_HEAD_MA ILING_ADDRESS1	7	C	35		RW	Postadress
ADK_AGREEMENT_HEAD_MA ILING_ADDRESS2	8	C	35		RW	Postadress 2
ADK_AGREEMENT_HEAD_ZIP CODE	9	C	12		RW	Postnummer
ADK_AGREEMENT_HEAD_CIT Y	10	C	24		RW	Ort
ADK_AGREEMENT_HEAD_CO UNTRY	11	C	24		RW	Land
ADK_AGREEMENT_HEAD_DE LIVERY_NAME	12	C	50		RW	Namn för leverans. Om blankt ska innehållet i fältet NAME användas.
ADK_AGREEMENT_HEAD_DE LIVERY_MAILING_ADDRESS1	13	C	35		RW	Postadress, gata, box för leverans. Om blankt ska innehållet i fältet MAILING_ADDRESS1 användas.

ADK_AGREEMENT_HEAD_DE LIVERY_MAILING_ADDRESS2	14	C	35		RW	Postadress, gata, box för leverans, rad 2. Om blankt ska innehållet i fältet MAILING_ADDRESS2 användas.
ADK_AGREEMENT_HEAD_DE LIVERY_ZIPCODE	15	C	12		RW	Postnr. Om blankt ska innehållet i fältet ZIPCODE användas.
ADK_AGREEMENT_HEAD_DE LIVERY_CITY	16	C	24		RW	Ortsnamn för leverans. Om blankt ska innehållet i fältet CITY användas.
ADK_AGREEMENT_HEAD_DE LIVERY_COUNTRY	17	C	24		RW	Land för leverans. Om blankt ska innehållet i fältet COUNTRY användas.
ADK_AGREEMENT_HEAD_CU STOMER_REFERENCE_NAME	18	C	50		RW	Er referens
ADK_AGREEMENT_HEAD_OU R_REFERENCE_NAME	19	C	25		RW	Vår referens
ADK_AGREEMENT_HEAD_DI STRICT_CODE	20	C	8		RW	Distrikt
ADK_AGREEMENT_HEAD_SE LLER_CODE	21	C	10		RW	Säljare
ADK_AGREEMENT_HEAD_CU RRENCY_CODE	22	C	4		RW	Valutakod
ADK_AGREEMENT_HEAD_LA NGUAGE_CODE	23	C	4		RW	Språkkod
ADK_AGREEMENT_HEAD_RE MINDER	24	L	1		RW	Kravbrev
ADK_AGREEMENT_HEAD_DE MAND_FEE	25	L	1		RW	Kravavgift
ADK_AGREEMENT_HEAD_IN VOICE_INTEREST	26	L	1		RW	Räntefaktura
ADK_AGREEMENT_HEAD_EX PORT	27	L	1		RW	Export
ADK_AGREEMENT_HEAD_IN CLUDING_VAT	28	L	1		RW	Inklusive moms
ADK_AGREEMENT_HEAD_LIS T_OF_PRICE_CODE	29	C	8		RW	Prislista
ADK_AGREEMENT_HEAD_IN VOICE_DISCOUNT	30	D	5	1	RW	Fakturarabatt
ADK_AGREEMENT_HEAD_CO DE_OF_DISCOUNT	31	C	16		RW	Kundrabattnummer
ADK_AGREEMENT_HEAD_CO DE_OF_TERMS_OF_PAYMEN T	32	C	8		RW	Betalningsvillkor
ADK_AGREEMENT_HEAD_CO DE_OF_TERMS_OF_DELIVER Y	33	C	8		RW	Leveransvillkor
ADK_AGREEMENT_HEAD_CO DE_OF_WAY_OF_DELIVERY	34	C	8		RW	Leveranssätt
ADK_AGREEMENT_HEAD_TE XT1	35	C	60		RW	Text 1
ADK_AGREEMENT_HEAD_TE XT2	36	C	60		RW	Text 2
ADK_AGREEMENT_HEAD_TE XT3	37	C	60		RW	Text 3
ADK_AGREEMENT_HEAD_SE T_OF_VAT1	38	D	8	5	RW	Moms 1
ADK_AGREEMENT_HEAD_SE T_OF_VAT2	39	D	8	5	RW	Moms 2
ADK_AGREEMENT_HEAD_SE T_OF_VAT3	40	D	8	5	RW	Moms 3
ADK_AGREEMENT_HEAD_SE T_OF_VAT4	41	D	8	5	RW	Moms 4
ADK_AGREEMENT_HEAD_CA RGO_AMOUNT	42	D	10	2	RW	Frakt
ADK_AGREEMENT_HEAD_CA RGO_VAT_CODE	43	C	1		RW	Frakt momskod
ADK_AGREEMENT_HEAD_DI SPATCH_FEE	44	D	10	2	RW	Expeditionsavgift
ADK_AGREEMENT_HEAD_VA T_CODE_DISPATCH_FEE	45	C	1		RW	Expeditionsavgift momskod
ADK_AGREEMENT_HEAD_RO UND_OFF	46	D	7	2	R	Avrundning
ADK_AGREEMENT_HEAD_TO TAL_AMOUNT	47	D	14	2	R	Total
ADK_AGREEMENT_HEAD_BA LANCE	48	D	14	2	R	Saldo

ADK_AGREEMENT_HEAD_PROJECT_CODE	49	C	10		RW	Projekt
ADK_AGREEMENT_HEAD_PROFIT_CENTRE	50	C	6		RW	Resultatenhet
ADK_AGREEMENT_HEAD_BASIS_OF_VAT0	51	D	14	2	R	Momsunderlag 0
ADK_AGREEMENT_HEAD_BASIS_OF_VAT1	52	D	14	2	R	Momsunderlag 1
ADK_AGREEMENT_HEAD_BASIS_OF_VAT2	53	D	14	2	R	Momsunderlag 2
ADK_AGREEMENT_HEAD_BASIS_OF_VAT3	54	D	14	2	R	Momsunderlag 3
ADK_AGREEMENT_HEAD_BASIS_OF_VAT4	55	D	14	2	R	Momsunderlag 4
ADK_AGREEMENT_HEAD_AMOUNT_OF_VAT1	56	D	14	2	R	Momsbelopp 1
ADK_AGREEMENT_HEAD_AMOUNT_OF_VAT2	57	D	14	2	R	Momsbelopp 2
ADK_AGREEMENT_HEAD_AMOUNT_OF_VAT3	58	D	14	2	R	Momsbelopp 3
ADK_AGREEMENT_HEAD_AMOUNT_OF_VAT4	59	D	14	2	R	Momsbelopp 4
ADK_AGREEMENT_HEAD_PENALTY_INTEREST_RATE	60	D	6	2	RW	Ränta
ADK_AGREEMENT_HEAD_CUSTOMER_DEMAND_ACCOUNT	61	C	6		RW	Kontonummer kundfordran
ADK_AGREEMENT_HEAD_CARGO_ACCOUNT	62	C	6		RW	Kontonummer frakt
ADK_AGREEMENT_HEAD_DISPATCH_FEE_ACCOUNT	63	C	6		RW	Kontonummer expeditionsavgift
ADK_AGREEMENT_HEAD_BALANCE2	64	D	14	2	R	Saldo kronor
ADK_AGREEMENT_HEAD_VAT1_ACCOUNT	65	C	6		RW	Kontonummer moms 1
ADK_AGREEMENT_HEAD_VAT2_ACCOUNT	66	C	6		RW	Kontonummer moms 2
ADK_AGREEMENT_HEAD_VAT3_ACCOUNT	67	C	6		RW	Kontonummer moms 3
ADK_AGREEMENT_HEAD_VAT4_ACCOUNT	68	C	6		RW	Kontonummer moms 4
ADK_AGREEMENT_HEAD_ROUND_OFF_ACCOUNT	69	C	6		RW	Kontonummer avrundning
ADK_AGREEMENT_HEAD_DISCOUNT_ACCOUNT	70	C	6		RW	Kontonummer given rabatt
ADK_AGREEMENT_HEAD_CUSTOMER_INVOICE_VAT_NUMBER	71	C	20		RW	Momsregistreringsnummer
ADK_AGREEMENT_HEAD_EU_ACCOUNT_FOR_QUARTERLY	72	L	1		RW	EU kvartalsrapport
ADK_AGREEMENT_HEAD_EU_THIRD_PART_TRADE	73	L	1		RW	EU trepartshandel
ADK_AGREEMENT_HEAD_SAVE_TEXT	74	L	1		RW	Spara text
ADK_AGREEMENT_HEAD_GROSS	75	D	14	2	R	Brutto
ADK_AGREEMENT_HEAD_NET	76	D	14	2	R	Netto
ADK_AGREEMENT_HEAD_EXCLUDING_OF_VAT	77	D	14	2	R	Exklusive moms
ADK_AGREEMENT_HEAD_VAT_AMOUNT	78	D	14	2	R	Momsbelopp
ADK_AGREEMENT_HEAD_CONTRIBUTION_MARGIN	79	D	14	2	R	Täckningbelopp
ADK_AGREEMENT_HEAD_VALUE	80	D	14	2	R	Värde
ADK_AGREEMENT_HEAD_TYPE_IN_TEXT	81	C	12		R	Typ
ADK_AGREEMENT_HEAD_ROWS	82	Data	1		RW	Rader
ADK_AGREEMENT_HEAD_ROWS	83	D	9	0	RW	Antal rader

ADK_AGREEMENT_HEAD_LO CAL_REMARK	84	C	120		RW	Kommentar (tidigare benämnt Lokal anmärkning)
ADK_AGREEMENT_HEAD_EA NLOC	85	C	20		RW	EAN lokaliseringsnr
ADK_AGREEMENT_HEAD_DE LIVERY_EANLOC	86	C	20		RW	EAN lokaliseringsnummer avvikande leveransadress
ADK_AGREEMENT_HEAD_DI SPATCHER	87	C	7		RW	Speditör
ADK_AGREEMENT_HEAD_CO DE_OF_COUNTRY	88	C	4		RW	Landskod
ADK_AGREEMENT_HEAD_DE LIVERY_CODE_OF_COUNTRY	89	C	4		RW	Lev landskod
ADK_AGREEMENT_HEAD_BU ILDSERVICE	90	L	1		RW	Byggtjänst
ADK_AGREEMENT_HEAD_TO T_SHIPWT	91	D	14	2	R	
ADK_AGREEMENT_HEAD_AG REEMENT_TYPE	92	C	4		R	Avtalstyp
ADK_AGREEMENT_HEAD_CO MPLETED	93	L	1		RW	Avtal avslutat
ADK_AGREEMENT_HEAD_DA TE_START	94	Date	8		RW	Avtal startdatum
ADK_AGREEMENT_HEAD_DA TE_END	95	Date	8		RW	Avtal slutdatum
ADK_AGREEMENT_HEAD_FIR ST_MONTH	96	D	4	0	RW	Första månad
ADK_AGREEMENT_HEAD_INT ERVAL	97	D	4	0	RW	Avtal intervall
ADK_AGREEMENT_HEAD_DA Y	98	D	4	0	RW	Avtal dag
ADK_AGREEMENT_HEAD_UP DATE_PRICE	99	L	1		RW	Avtal prisuppdatering
ADK_AGREEMENT_HEAD_UP DATE_ADRESS	100	L	1		RW	Avtal adressuppdatering
ADK_AGREEMENT_HEAD_PE RIOD	101	L	1		RW	Avtal period
ADK_AGREEMENT_HEAD_AV ERAGE_UNIT_PRICE_DONT_ UPDATE	102	L	1		RW	Uppdatera ej kalkylpris
ADK_AGREEMENT_HEAD_NE XT_INVOICE	103	Date	8		R	Avtal nästa faktura
ADK_AGREEMENT_HEAD_PE RIOD_START	104	Date	8		RW	Avtal periodstart
ADK_AGREEMENT_HEAD_PE RIOD_END	105	Date	8		RW	Avtal periodslut
ADK_AGREEMENT_HEAD_PR ICE_CHANGE	106	Date	8		R	Avtal prisändring datum
ADK_AGREEMENT_HEAD_AG REEMENT_INTERVAL	107	C	24		R	Avtal avtalstid
ADK_AGREEMENT_HEAD_NE XT_PERIOD_INTERVAL	108	C	24		R	Avtal nästa period
ADK_AGREEMENT_HEAD_AC TIVE_YEAR	109	L	1		RW	Avtal aktivt år
ADK_AGREEMENT_HEAD_CU RRENCY_RATE	110	D	12	6	RW	Valutakurs
ADK_AGREEMENT_HEAD_CU RRENCY_UNIT	111	D	9	0	R	Valutaenhet
ADK_AGREEMENT_HEAD_TI MESTAMP	112	Date	10		R	Tidsstämpel
ADK_AGREEMENT_HEAD_TA X_REDUCTION_INCLUDED	113	L	1		RW	Innehåller husarbete
ADK_AGREEMENT_HEAD_TA X_REDUCTION_TAX_BASIS	114	D	14	2	R	Undelag för skattereduktion
ADK_AGREEMENT_HEAD_TA X_REDUCTION_NUMBER_OF _PERSONS	115	D	4	0	R	Antal personer (fördelad skattereduktion)
ADK_AGREEMENT_HEAD_TA X_REDUCTION_BUILDING_W ORK	116	L	1		RW	Avtalet avser ROT-arbete
ADK_AGREEMENT_HEAD_TA X_REDUCTION_HOUSING_CO OPERATIVE_IDENTITY_NUMB ER	117	C	14		RW	Bostadsrättsföreningens organisationsnummer

ADK_AGREEMENT_HEAD_TAX_REDUCTION_DESCRIPTOR_OF_PROPERTY	118	C	100		RW	Fastighetsbeteckning
ADK_AGREEMENT_HEAD_ELECTRONIC_REFERENCE	119	C	35		RW	Referenskod

Avtalsrader – ADK_DB_AGREEMENT_ROW

Program: Visma Administration 1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_OOI_ROW_ARTICLE_NUMBER	0	C	40		RW	Artikelnummer
ADK_OOI_ROW_SUPPLIER_ARTICLE_NUMBER	1	C	40		R	Leverantörens artikelnummer. Används ej i avtal.
ADK_OOI_ROW_QUANTITY1	2	D	12	4	RW	Antalsfält. Avtal: Levererat antal.
ADK_OOI_ROW_QUANTITY2	3	D	12	4	RW	Extra antalsfält. Avtal: Används ej.
ADK_OOI_ROW_QUANTITY3	4	D	12	4	RW	Extra antalsfält. Avtal: Används ej.
ADK_OOI_ROW_PRICE_EACH_CURRENT_CURRENCY	5	D	12	4/2	RW	Pris per enhet i aktuell valuta. Inklusive eller exklusive moms.
ADK_OOI_ROW_PRICE2	6	D	12	4/2	RW	Extra fält för pris. Avtal: Används ej
ADK_OOI_ROW_DISCOUNT	7	D	10	2	RW	Rabatt i procent eller belopp. Flagga indikerar vilket. Alltid positivt.
ADK_OOI_ROW_AMOUNT_DISCOUNT_FLAG	8	L	1		RW	Sann indikerar beloppsrabatt, annars procentrabatt. Alltid positivt.
ADK_OOI_ROW_AMOUNT_CURRENT_CURRENCY	9	D	14	2	RW	Totalt belopp i aktuell valuta inklusive eller exklusive moms. Normalt positivt.
ADK_OOI_ROW_AMOUNT_DOMESTIC_CURRENCY	10	D	14	2	R	Totalt belopp i inhemsk valuta utan moms. Normalt positivt.
ADK_OOI_ROW_ACCOUNT_NUMBER	11	C	6		RW	Kontonummer.
ADK_OOI_ROW_PROFIT_CENTRE	12	C	6		RW	Resultatenhet. Definieras i eget register.
ADK_OOI_ROW_PROJECT (4)	13	C	10		RW	Projekt. Definieras i eget register.
ADK_OOI_ROW_TEXT	14	C	60		RW	Används för text. Exempelvis artikelbenämning i offert, order, faktura och avtal.
ADK_OOI_ROW_UNIT	15	C	4		RW	Enhetskod. Definieras i eget register.
ADK_OOI_ROW_VAT_CODE	16	C	1		RW	Momskod. Påverkas inte om inkl moms eller ej.
ADK_OOI_ROW_NOTE_OF_OUTSTANDING_ORDERS (1)	17	L	1		RW	Används i order för att indikera att differens mellan beställt och levererat inte ska restnoteras.
ADK_OOI_ROW_DATE2	18	Date	8		RW	Extra datumfält. Används i order för beräknat leveransdatum och i faktura för restnoterat leveransdatum.
ADK_OOI_ROW_TYPE_OF_ROW	19	C	1		RW	Radtyp. Användning beror på typ, normalt blank. 'T' = Textrad, 'D' – deltotal (på samlingsfaktura rad med ordersumma) 'R' – Ränterad.
ADK_OOI_ROW_CONTRIBUTION_DEGREE	20	D	6	1	R	Täckningsgrad.
ADK_OOI_ROW_CONTRIBUTION_MARGIN	21	D	14	2	R	Täckningsbidrag. Blankas om noll.
ADK_OOI_ROW_PRINT	22	L	1		RW	Används inte i avtal
ADK_OOI_ROW_PRINT2	23	L	1		RW	Används inte i avtal
	24	U	0		R	Används inte
ADK_OOI_ROW_BAR_CODE	25	C	32		RW	Används inte i avtal
ADK_OOI_ROW_DATE3	26	Date	8		R	Används inte i avtal
ADK_OOI_ROW_ARTICLE_TYPE (5)	27	C	1		R	Artikeltyp
ADK_OOI_ROW_FROM_TYPE	28	C	1		R	Används inte i avtal
ADK_OOI_ROW_FROM_DOCUMENT	29	D	15	0	R	Används inte i avtal
ADK_OOI_ROW_FROM_DOCUMENT	30	D	9	0	R	Används inte i avtal
ADK_OOI_ROW_ROWNUMBER	31	D	9	0	R	Radnummer
ADK_OOI_ROW_CONNECTION_TYPE	32	C	3		R	Används inte i avtal
ADK_OOI_ROW_CONNECTION_DOCUMENT	33	D	15	0	R	Används inte i avtal
ADK_OOI_ROW_CONNECTION_DOCUMENT	34	D	9	0	R	Används inte i avtal

ADK_OOI_ROW_SERIAL_NUM BER	35	C	30		RW	Används inte i avtal
ADK_OOI_ROW_BATCH_NUM BER	36	C	30		RW	Används inte i avtal
ADK_OOI_ROW_BEST_BEFOR E	37	Date	8		RW	Används inte i avtal
ADK_OOI_ROW_NOTIFY_QT1	38	D	12	4	R	Används inte i avtal
ADK_OOI_ROW_NOTIFY_QT2	39	D	12	4	R	Används inte i avtal
ADK_OOI_ROW_PIN_BOOKIN G	40	L	1		R	Används inte i avtal
ADK_OOI_ROW_NUM_PIN_BO OKING	41	D	9	0	R	Används inte i avtal
ADK_OOI_ROW_LABOUR_CO ST	42	L	1		RW	Arbetskostnad
ADK_OOI_ROW_QT_BOOKING	43	D	12	3	R	Används inte i avtal
ADK_OOI_ROW_ORDER_BAS E_TYPE	44	C	1		R	Används inte i avtal
ADK_OOI_ROW_USER_ID	45	C	15		R	Medarbetarid
ADK_OOI_ROW_STRUCTURE _ORDER (1)	46	D	9	0	R	Ordning i struktur
ADK_OOI_ROW_PACKAGE (1)	47	L	1		R	Paket
ADK_OOI_ROW_STRUCTURE (1)	48	L	1		R	Struktur
ADK_OOI_ROW_TIMESTAMP	49	Date	10		R	Tidsstämpel
ADK_OOI_ROW_TAX_REDUCE TION_TYPES_ID (2)	50	C	4		RW	Husarbetstypid.
ADK_OOI_ROW_HOURS_WOR KED	51	D	7	1	R	Används inte i avtal.
ADK_OOI_ROW_H_SCHAB_P ROC (2)	52	D	6	2	RW	Schablonprocentsats för vissa husarbetstyper.

(1) Fältet används endast i Visma Administration 2000.

(2) Fältet används endast i Visma Administration 1000/2000.

(4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

(5) Fältet används endast i Visma Förening.

Leverantörsfakturor – ADK_DB_SUPPLIER_INVOICE_HEAD

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_SUP_INV_HEAD_GIVEN _NUMBER	0	D	15	0	RW	Unikt löpnummer i egen nummerserie.
ADK_SUP_INV_HEAD_SUPPLI ER_NUMBER	1	C	16		RW	Leverantörsnummer.
ADK_SUP_INV_HEAD_SUPPLI ER_NAME	2	C	50		RW	Leverantörens namn.
ADK_SUP_INV_HEAD_PGIRO	3	C	13		RW	Plusgiro.
ADK_SUP_INV_HEAD_ORGA NISATION_NUMBER	4	C	25		RW	OCR kod som är krav på vissa plusgiroavier.
ADK_SUP_INV_HEAD_BGIRO	5	C	34		RW	Bankgiro.
ADK_SUP_INV_HEAD_INVOIC E_NUMBER	6	C	25		RW	Fakturanr.
ADK_SUP_INV_HEAD_CUSTO MER_NUMBER	7	C	20		RW	Vårt kundnummer hos leverantör.
ADK_SUP_INV_HEAD_INVOIC E_DATE	8	Date	8		RW	Fakturans datum.
ADK_SUP_INV_HEAD_DUE_D ATE	9	Date	8		RW	Förfallodatum.
ADK_SUP_INV_HEAD_CURRE NCY_CODE (3)	10	C	4		RW	Valutakod.
ADK_SUP_INV_HEAD_CURRE NCY_RATE (3)	11	D	12	6	RW	Valutakurs.
ADK_SUP_INV_HEAD_TYPE_ OF_INVOICE	12	C	1		RW	Fakturatyp: F= Vanlig faktura, K=Kreditfaktura, R= Räntefaktura. . Endast möjligt att skapa vanliga fakturor och kreditfakturor.
ADK_SUP_INV_HEAD_TOTAL	13	D	14	2	RW	Fakturans belopp att betala (i fakturans valuta).

ADK_SUP_INV_HEAD_BALANCE_CURRENT_CURRENCY	14	D	14	2	R	Fakturans saldo i fakturans valuta. Betalningar på väg ingår inte i detta fält. I kreditfakturor är detta belopp negativt.
ADK_SUP_INV_HEAD_OUTGOING_PAYMENTS_CURRENT_CURRENCY	15	D	14	2	R	Betalningar på väg (i inhemsk valuta).
ADK_SUP_INV_HEAD_VAT_AMOUNT	16	D	14	2	RW	Momsbelopp.
ADK_SUP_INV_HEAD_CARGO_FEE	17	D	10	2	RW	Frakt (i fakturans valuta). I kreditfakturor är detta belopp negativt.
ADK_SUP_INV_HEAD_NAME_OF_CERTIFIER	18	C	20		RW	Namn på den som ska attestera.
ADK_SUP_INV_HEAD_CERTIFIED	19	L	1		RW	Sann betyder att fakturan är attesterad.
ADK_SUP_INV_HEAD_CANCELLED	20	L	1		RW	Makulerad.
ADK_SUP_INV_HEAD_PROJECT (4)	21	C	10		RW	Projekt. Definieras i eget register.
ADK_SUP_INV_HEAD_PROFIT_CENTRE	22	C	6		RW	Resultatenhet. Definieras i eget register.
ADK_SUP_INV_HEAD_JOURNAL_NUMBER	23	D	15	0	R	Journalnummer.
ADK_SUP_INV_HEAD_NOTE1	24	C	60		RW	Fritt anteckningsfält.
ADK_SUP_INV_HEAD_NOTE2	25	C	60		RW	Fritt anteckningsfält.
ADK_SUP_INV_HEAD_NOTE3	26	C	60		RW	Fritt anteckningsfält.
ADK_SUP_INV_HEAD_CURRENCY_UNIT (3)	27	D	9	0	R	Valutaenhet. Definieras i eget register.
ADK_SUP_INV_HEAD_SUPPLIER_DEBT_ACCOUNT	28	C	6		RW	Konto för skulder till leverantör.
ADK_SUP_INV_HEAD_CODE_OF_CENTRAL_BANK (3)	29	C	3		RW	Riksbankskod
ADK_SUP_INV_HEAD_PAYWITH_CURRENCY_ACCOUNT (3)	30	L	1		RW	Sann: Betala med valutakonto
ADK_SUP_INV_HEAD_PAYMENT_ABROAD (3)	31	L	1		R	Sann: Utlandsbetalning
ADK_SUP_INV_HEAD_CODE_OF_CHARGE_ON_PAYMENT_ABROAD (3)	32	D	4	0	RW	Avgiftskod bankbetalning utland. Tillåtna värden, se ADK_CODE_BG_FOREIGN_FEE i adk.h.
ADK_SUP_INV_HEAD_TYPE_INFORMATION_TEXT	33	C	8		R	Fakturans typ som klartext. "Faktura", "Kredit" eller "Ränta".
ADK_SUP_INV_HEAD_VERIFICATION_SERIES_ID	34	C	10		R	Verifikationsserie + Verifikationsnummer
ADK_SUP_INV_HEAD_CHANNEL_OF_STOCK (2)	35	L	1		RW	Sann betyder att inleverans görs via denna faktura. Ska inte kunna sättas om följesedel redan registrerats.
ADK_SUP_INV_HEAD_ROWS	36	Data	1		RW	Pekar ut tillhörande rader
ADK_SUP_INV_HEAD_NROWS	37	D	9	0	RW	Antalet tillhörande rader
ADK_SUP_INV_HEAD_CREDIT_INVOICE_COPY_NUMBER	38	D	15	0	RW	Faktura att kreditera vid skapande av kreditfaktura.
ADK_SUP_INV_HEAD_CREDIT_INVOICE_COPY_WHAT	39	D	1	0	RW	Vad som ska kopieras vid skapande av kreditfaktura. Se ADK_CREDIT_INVOICE_TO_COPY för specification i adk.h.
ADK_SUP_INV_HEAD_CREDIT_NUMBER	40	C	10		R	Den faktura som fakturan krediterar eller fakturan som krediteras vid kreditfaktura.
ADK_SUP_INV_HEAD_IGNORE_CHECK_INVNR	41	L	1		RW	Används för att ignorera kontroll på fältet Fakturanr
ADK_SUP_INV_HEAD_DOCUMENT	42	C	160		RW	Dokument
ADK_SUP_INV_HEAD_TIMESTAMP	43	Date	10		R	Tidsstämpel
ADK_SUP_INV_HEAD_AUTOGIRO	44	L	1		RW	Autogiro
ADK_SUP_INV_HEAD_PAYMSTOP	45	L	1		RW	Betalspär
ADK_SUP_INV_HEAD_LOCAL_REMARK	46	C	120		RW	Kommentar
ADK_SUP_INV_HEAD_IMPORTED_XML_NAME	47	C	60		R	Filnamn för xmlfil när e-fakturan är importerad. Filen finns i katalogen SPCXML_IMPORTKIV i företagskatalogen.

- (2) Fältet används endast i Visma Administration 1000/2000.
 (3) Fältet används endast i Visma Administration 500/1000/2000.
 (4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

Leverantörsfakturarader – ADK_DB_SUPPLIER_INVOICE_ROW

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_OOI_ROW_ARTICLE_NUMBER (2)	0	C	40		RW	Artikelnummer
ADK_OOI_ROW_SUPPLIER_ARTICLE_NUMBER (2)	1	C	40		RW	Leverantörens artikelnummer.
ADK_OOI_ROW_QUANTITY1 (2)	2	D	12	4	RW	Antalsfält. Leverantörsfaktura: Levererat antal
ADK_OOI_ROW_QUANTITY2 (1)	3	D	12	4	RW	Extra antalsfält. Leverantörsfaktura: Restnoterat.
ADK_OOI_ROW_QUANTITY3	4	D	12	4	R	Extra antalsfält. Leverantörsfaktura: Väntas.
ADK_OOI_ROW_PRICE_EACH_CURRENT_CURRENCY (2)	5	D	12	4/2	R	Pris per enhet i aktuell valuta. Inklusive eller exklusive moms.
ADK_OOI_ROW_PRICE2 (2)	6	D	12	4/2	R	Extra fält för pris. I offert, order, och faktura används detta fält för kalkylpris per enhet utan moms. Används för att visa beräknat täckningsbidrag.
ADK_OOI_ROW_DISCOUNT (2)	7	D	10	2	R	Rabatt i procent eller belopp. Flagga indikerar vilket. Alltid positivt.
ADK_OOI_ROW_AMOUNT_DISCOUNT_FLAG (2)	8	L	1		R	Sann indikerar beloppsrabatt, annars procentrabatt. Alltid positivt.
ADK_OOI_ROW_AMOUNT_CURRENT_CURRENCY (3)	9	D	14	2	RW	Totalt belopp i aktuell valuta inklusive eller exklusive moms. Normalt positivt.
ADK_OOI_ROW_AMOUNT_DOMESTIC_CURRENCY	10	D	14	2	RW	Totalt belopp i inhemsk valuta utan moms. Normalt positivt.
ADK_OOI_ROW_ACCOUNT_NUMBER	11	C	6		RW	Kontonummer.
ADK_OOI_ROW_PROFIT_CENTRE	12	C	6		RW	Resultatenhet. Definieras i eget register.
ADK_OOI_ROW_PROJECT (4)	13	C	10		RW	Projekt. Definieras i eget register.
ADK_OOI_ROW_TEXT	14	C	60		RW	Används för text. Exempelvis artikelbenämning i offert, order, faktura.
ADK_OOI_ROW_UNIT (2)	15	C	4		RW	Enhetskod. Definieras i eget register.
ADK_OOI_ROW_VAT_CODE	16	C	1		R	Används inte i leverantörsfaktura.
ADK_OOI_ROW_NOTE_OF_OUTSTANDING_ORDERS (1)	17	L	1		R	Används i order för att indikera att differens mellan beställt och levererat inte ska restnoteras.
ADK_OOI_ROW_DATE2 (2)	18	Date	8		R	Extra datumfält. Används i order för beräknat leveransdatum och i faktura för restnoterat leveransdatum.
ADK_OOI_ROW_TYPE_OF_ROW (2)	19	C	1		R	Radtyp. Användning beror på typ, normalt blank. 'T' = Textrad, 'D' – deltotal (på samlingsfaktura rad med ordersumma) 'R' – Ränterad.
ADK_OOI_ROW_CONTRIBUTION_DEGREE (2)	20	D	6	1	R	Täckningsgrad.
ADK_OOI_ROW_CONTRIBUTION_MARGIN (2)	21	D	14	2	R	Täckningsbidrag. Blankas om noll.
ADK_OOI_ROW_PRINT	22	L	1		R	Används inte i Levfaktura
ADK_OOI_ROW_PRINT2	23	L	1		R	Används inte i Levfaktura
	24	U	0		R	Används inte
ADK_OOI_ROW_BAR_CODE (2)	25	C	32		RW	Streckkod.
ADK_OOI_ROW_DATE3	26	Date	8		R	Används inte i Levfaktura
ADK_OOI_ROW_ARTICLE_TYPE	27	C	1		R	Används inte i Levfaktura
ADK_OOI_ROW_FROM_TYPE	28	C	1		RW	Dokumenttyp
ADK_OOI_ROW_FROM_DOCUMENT	29	D	15	0	RW	Dokumentnummer
ADK_OOI_ROW_FROM_DOCUMENT	30	D	9	0	RW	Dokumentrad
ADK_OOI_ROW_ROWNUMBER	31	D	9	0	R	Radnummer

ADK_OOI_ROW_CONNECTION_TYPE	32	C	3		R	Kopplingstyp
ADK_OOI_ROW_CONNECTION_DOCUMENT	33	D	15	0	R	Kopplingsdokument
ADK_OOI_ROW_CONNECTION_DOCROW	34	D	9	0	R	Kopplingsnummer
ADK_OOI_ROW_SERIAL_NUMBER	35	C	30		R	Används inte i leverantörsfaktura
ADK_OOI_ROW_BATCH_NUMBER	36	C	30		R	Används inte i leverantörsfaktura
ADK_OOI_ROW_BEST_BEFORE	37	Date	8		R	Används inte i leverantörsfaktura
ADK_OOI_ROW_NOTIFY_QT1	38	D	12	4	R	Används inte i leverantörsfaktura
ADK_OOI_ROW_NOTIFY_QT2	39	D	12	4	R	Används inte i leverantörsfaktura
ADK_OOI_ROW_PIN_BOOKING	40	L	1		R	Används inte i leverantörsfaktura
ADK_OOI_ROW_NUM_PIN_BOOKING	41	D	9	0	R	Används inte i leverantörsfaktura
ADK_OOI_ROW_LABOUR_COST	42	L	1		RW	Används inte i leverantörsfaktura
ADK_OOI_ROW_QT_BOOKING	43	D	12	3	R	Kvantitet (används av leverantörsfaktura)
ADK_OOI_ROW_ORDER_BASE_TYPE	44	C	1		R	Används inte i leverantörsfaktura
ADK_OOI_ROW_USER_ID	45	C	15		R	Medarbetarid
ADK_OOI_ROW_STRUCTURE_ORDER	46	D	9	0	R	Används inte i leverantörsfaktura
ADK_OOI_ROW_PACKAGE	47	L	1		R	Används inte i leverantörsfaktura
ADK_OOI_ROW_STRUCTURE	48	L	1		R	Används inte i leverantörsfaktura
ADK_OOI_ROW_TIMESTAMP	49	Date	10		R	Tidsstämpel
ADK_OOI_ROW_TAX_REDUCTION_TYPES_ID	50	C	4		R	Används inte i levfaktura.
ADK_OOI_ROW_HOURS_WORKED	51	D	7	1	R	Används inte i levfaktura.
ADK_OOI_ROW_H_SCHAB_PROC	52	D	6	2	R	Används inte i levfaktura

(1) Fältet används endast i Visma Administration 2000.

(2) Fältet används endast i Visma Administration 1000/2000.

(3) Fältet används endast i Visma Administration 500/1000/2000.

(4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

Beställning – ADK_DB_BOOKING_HEAD

Program: Visma Administration 2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_BOOKING_DOCUMENT_NUMBER	0	D	15	0	RW	Beställningsnummer
ADK_BOOKING_SUPPLIER_NUMBER	1	C	16		RW	Leverantörsnr
ADK_BOOKING_SUPPLIER_NAME	2	C	50		RW	Leverantörens namn
ADK_BOOKING_MAILING_ADDRESS	3	C	35		RW	Leverantörens adress
ADK_BOOKING_MAILING_ADDRESS2	4	C	35		RW	Leverantörens adress, rad 2
ADK_BOOKING_ZIPCODE	5	C	12		RW	Leverantörens postnr
ADK_BOOKING_CITY	6	C	24		RW	Leverantörens ort
ADK_BOOKING_COUNTRY	7	C	24		RW	Leverantörens land
ADK_BOOKING_DELIVERY_NAME	8	C	50		RW	Avvikande leveransadress – namn
ADK_BOOKING_DELIVERY_MAILING_ADDRESS1	9	C	35		RW	Avvikande leveransadress – adress
ADK_BOOKING_DELIVERY_MAILING_ADDRESS2	10	C	35		RW	Avvikande leveransadress – adress, rad 2
ADK_BOOKING_VISITING_ADDRESS	11	C	35		RW	Avvikande leveransadress – besöksadress
ADK_BOOKING_DELIVERY_ZIPCODE	12	C	12		RW	Avvikande leveransadress – postnr

ADK_BOOKING_DELIVERY_CITY	13	C	24		RW	Avvikande leveransadress – ort
ADK_BOOKING_DELIVERY_COUNTRY	14	C	24		RW	Avvikande leveransadress – land
ADK_BOOKING_SUPPLIER_CUSTOMER_REFERENCE_NAME	15	C	25		RW	Leverantörens referens
ADK_BOOKING_SUPPLIER_REFERENCE	16	C	50		RW	Er referens
ADK_BOOKING_OUR_REFERENCE	17	C	25		RW	Vår referens
ADK_BOOKING_LANGUAGE_CODE	18	C	4		RW	Språkkod
ADK_BOOKING_CURRENCY_CODE	19	C	4		RW	Valutakod
ADK_BOOKING_SUPPLIER_ORDER_NUMBER	20	C	16		RW	Leverantörens ordernummer
ADK_BOOKING_NOT_DONE	21	L	1		RW	Ej klar
ADK_BOOKING_DELIVERED	22	L	1		R	Levererad
ADK_BOOKING_CANCELLED	23	L	1		RW	Makulerad
ADK_BOOKING_PRINTED	24	L	1		RW	Utskriven
ADK_BOOKING_CODE_OF_TERMS_OF_PAYMENT	25	C	8		RW	Betalningsvillkor
ADK_BOOKING_CODE_OF_TERMS_OF_DELIVERY	26	C	8		RW	Leveransvillkor
ADK_BOOKING_CODE_OF_WAY_OF_DELIVERY	27	C	8		RW	Leveranssätt
ADK_BOOKING_DATE	28	Date	8		RW	Beställningsdatum
ADK_BOOKING_DELIVERY_DATE	29	Date	8		RW	Leveransdatum
ADK_BOOKING_PROJECT_CODE	30	C	10		RW	Projekt
ADK_BOOKING_PROFIT_CENTRE	31	C	6		RW	Resultatenhet
ADK_BOOKING_TOTAL_AMOUNT	32	D	14	2	R	Total
ADK_BOOKING_TEXT1	33	C	60		RW	Text1
ADK_BOOKING_TEXT2	34	C	60		RW	Text2
ADK_BOOKING_TEXT3	35	C	60		RW	Text3
ADK_BOOKING_OUR_CUSTOMER_NUMBER	36	C	20		RW	Vårt kundnr
ADK_BOOKING_EANLOC	37	C	20		RW	EAN lokaliseringssnr
ADK_BOOKING_DELIVERY_EANLOC	38	C	20		RW	EAN lokaliseringsnummer avvikande leveransadress
ADK_BOOKING_ROWS	39	Data	1		RW	Rader
ADK_BOOKING_NROWS	40	D	9	0	RW	Antal rader
ADK_BOOKING_CODE_OF_COUNTRY	41	C	4		RW	Landskod
ADK_BOOKING_DELIVERY_CODE_OF_COUNTRY	42	C	4		RW	Landskod avvikande leveransadress
ADK_BOOKING_CONTRACTNUMBER	43	C	15		RW	Kontraksnummer
ADK_BOOKING_TIMESTAMP	44	Date	10		R	Tidsstämpel
ADK_BOOKING_PRINTOPTION (7)	45	D	4	0	RW	Utskriftsval dokumentet
ADK_BOOKING_ELECTRONIC_REFERENCE	46	C	35		RW	Elektronisk referens
ADK_BOOKING_ELECTRONIC_ATTACHMENT	47	C	160		RW	Elektronisk bilaga
ADK_BOOKING_LOCAL_REMARK	48	C	120		RW	Kommentar

(7) För tillåtna värden läs avsnitt Utskriftsval i kunder, leverantörer och tillhörande dokument ovan

Beställningsrader – ADK_DB_BOOKING_ROW

Program: Visma Administration 2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
----------	----	-----	-------	-----	------	-------------

ADK_OOI_ROW_ARTICLE_NUMBER	0	C	40		RW	Artikelnummer
ADK_OOI_ROW_SUPPLIER_ARTICLE_NUMBER	1	C	40		RW	Leverantörens artikelnummer.
ADK_OOI_ROW_QUANTITY1	2	D	12	4	RW	Antalsfält. Leverantörsfaktura: Levererat antal. Beställt.
ADK_OOI_ROW_QUANTITY2	3	D	12	4	RW	Extra antalsfält. Leverantörsfaktura: Restnoterat. Väntas.
ADK_OOI_ROW_QUANTITY3	4	D	12	4	R	Extra antalsfält. Leverantörsfaktura: Väntas. Levererade.
ADK_OOI_ROW_PRICE_EACH_CURRENT_CURRENCY	5	D	12	2	RW	Pris per enhet i aktuell valuta. Inklusive eller exklusive moms.
ADK_OOI_ROW_PRICE2	6	D	12	2	R	Extra fält för pris. I offert, order, och faktura används detta fält för kalkylpris per enhet utan moms. Används för att visa beräknat täckningsbidrag.
ADK_OOI_ROW_DISCOUNT	7	D	10	2	R	Rabatt i procent eller belopp. Flagga indikerar vilket. Alltid positivt.
ADK_OOI_ROW_AMOUNT_DISCOUNT_FLAG	8	L	1		R	Sann indikerar beloppsrabatt, annars procentrabatt. Alltid positivt.
ADK_OOI_ROW_AMOUNT_CURRENT_CURRENCY	9	D	14	2	RW	Totalt belopp i aktuell valuta inklusive eller exklusive moms. Normalt positivt.
ADK_OOI_ROW_AMOUNT_DOMESTIC_CURRENCY	10	D	14	2	R	Totalt belopp i inhemsk valuta utan moms. Normalt positivt.
ADK_OOI_ROW_ACCOUNT_NUMBER	11	C	6		R	Kontonummer.
ADK_OOI_ROW_PROFIT_CENTRE	12	C	6		RW	Resultatenhet. Definieras i eget register.
ADK_OOI_ROW_PROJECT	13	C	10		RW	Projekt. Definieras i eget register.
ADK_OOI_ROW_TEXT	14	C	60		RW	Används för text. Exempelvis artikelbenämning i offert, order, faktura.
ADK_OOI_ROW_UNIT	15	C	4		RW	Enhetskod. Definieras i eget register.
ADK_OOI_ROW_VAT_CODE	16	C	1		R	Används inte i beställning.
ADK_OOI_ROW_NOTE_OF_OUTSTANDING_ORDERS	17	L	1		R	Används i order för att indikera att differens mellan beställt och levererat inte ska restnoteras.
ADK_OOI_ROW_DATE2	18	Date	8		RW	Extra datumfält. Används i order för beräknat leveransdatum och i faktura för restnoterat leveransdatum.
ADK_OOI_ROW_TYPE_OF_ROW	19	C	1		R	Radtyp. Användning beror på typ, normalt blank. 'T'= Textrad, 'D' – deltotal (på samlingsfaktura rad med ordersumma) 'R' – Rånträd.
ADK_OOI_ROW_CONTRIBUTION_DEGREE	20	D	6	1	R	Täckningsgrad.
ADK_OOI_ROW_CONTRIBUTION_MARGIN	21	D	14	2	R	Täckningsbidrag. Blankas om noll.
ADK_OOI_ROW_PRINT	22	L	1		R	Används inte i Beställning
ADK_OOI_ROW_PRINT2	23	L	1		R	Används inte i Beställning
	24	U	0		R	Används inte
ADK_OOI_ROW_BAR_CODE	25	C	32		RW	Streckkod.
ADK_OOI_ROW_DATE3	26	Date	8		R	Används inte i Beställning
ADK_OOI_ROW_ARTICLE_TYPE	27	C	1		R	Används inte i Beställning
ADK_OOI_ROW_FROM_TYPE	28	C	1		RW	Dokumenttyp
ADK_OOI_ROW_FROM_DOCUMENT	29	D	15	0	RW	Dokumentnummer
ADK_OOI_ROW_FROM_DOCUMENT	30	D	9	0	RW	Dokumentrad
ADK_OOI_ROW_ROWNUMBER	31	D	9	0	R	Radnummer
ADK_OOI_ROW_CONNECTION_TYPE	32	C	3		R	Kopplingstyp
ADK_OOI_ROW_CONNECTION_DOCUMENT	33	D	15	0	R	Kopplingsdokument
ADK_OOI_ROW_CONNECTION_DOCUMENT	34	D	9	0	R	Kopplingsnummer
ADK_OOI_ROW_SERIAL_NUMBER	35	C	30		R	Används inte i beställning
ADK_OOI_ROW_BATCH_NUMBER	36	C	30		R	Används inte i beställning
ADK_OOI_ROW_BEST_BEFORE	37	Date	8		R	Används inte i beställning
ADK_OOI_ROW_NOTIFY_QT1	38	D	12	4	R	Används inte i beställning

ADK_OOI_ROW_NOTIFY_QT2	39	D	12	4	R	Används inte i beställning
ADK_OOI_ROW_PIN_BOOKING	40	L	1		R	Häftad textrad
ADK_OOI_ROW_NUM_PIN_BOOKING	41	D	9	0	R	Antalet häftade textrader
ADK_OOI_ROW_LABOUR_COST	42	L	1		RW	Används inte i beställning
ADK_OOI_ROW_QT_BOOKING	43	D	12	3	R	Kvantitet (används av leverantörsfaktura)
ADK_OOI_ROW_ORDER_BASE_TYPE	44	C	1		R	Används inte i beställning
ADK_OOI_ROW_USER_ID	45	C	15		R	Medarbetarid
ADK_OOI_ROW_STRUCTURE_ORDER	46	D	9	0	R	Används inte i beställning
ADK_OOI_ROW_PACKAGE	47	L	1		R	Används inte i beställning
ADK_OOI_ROW_STRUCTURE	48	L	1		R	Används inte i beställning
ADK_OOI_ROW_TIMESTAMP	49	Date	10		R	Tidsstämpel
ADK_OOI_ROW_TAX_REDUCTION_TYPES_ID	50	C	4		R	Används inte i beställning.
ADK_OOI_ROW_HOURS_WORKED	51	D	7	1	R	Används inte i beställning.
ADK_OOI_ROW_H_SCHAB_PROC	52	D	6	2	R	Används inte i beställning.

Inkommande följesedel – ADK_DB_DELIVERY_NOTE_HEAD

Program: Visma Administration 2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_DELIVERY_NOTE_DOCUMENT_NUMBER	0	D	15	0	RW	Dokumentnummer
ADK_DELIVERY_NOTE_SUPPLIER_NUMBER	1	C	16		RW	Leverantörsnr
ADK_DELIVERY_NOTE_SUPPLIER_NAME	2	C	50		RW	Leverantörens namn
ADK_DELIVERY_NOTE_DELIVERY_NUMBER	3	C	16		RW	Inkommande följesedelsnummer
ADK_DELIVERY_NOTE_DATE	4	Date	8		RW	Följesedelsdatum
ADK_DELIVERY_NOTE_ARRIVAL_DATE	5	Date	8		RW	Ankomstdatum
ADK_DELIVERY_NOTE_CANCELLED	6	L	1		RW	Makulerad
ADK_DELIVERY_NOTE_NOTE1	7	C	60		RW	Anmärkning 1
ADK_DELIVERY_NOTE_NOTE2	8	C	60		RW	Anmärkning 2
ADK_DELIVERY_NOTE_NOTE3	9	C	60		RW	Anmärkning 3
ADK_DELIVERY_NOTE_PROJECT_CODE	10	C	10		RW	Projekt
ADK_DELIVERY_NOTE_PROFIT_CENTRE	11	C	6		RW	Resultatenhet
ADK_DELIVERY_NOTE_INVOICE_SENT	12	L	1		R	Fakturerad
ADK_DELIVERY_NOTE_CURRENCY_CODE	13	C	4		R	Valutakod
ADK_DELIVERY_NOTE_ROWS	14	Data	1		RW	Rader
ADK_DELIVERY_NOTE_ROWS	15	D	9	0	RW	Antal rader
ADK_DELIVERY_NOTE_DN_ARRIVAL_DATE	16	Date	8		RW	Leveransaviseringsankomstdatum
ADK_DELIVERY_NOTE_DN_DELIVER_DATE	17	Date	8		RW	Leveransaviseringsdatum
ADK_DELIVERY_NOTE_DN_NUMBER	18	C	16		RW	Leveransaviseringsnummer
ADK_DELIVERY_NOTE_DN_NOT_DONE	19	L	1		RW	Ej klar

ADK_DELIVERY_NOTE_DN_CREATED_ED	20	L	1		R	Skapad av EDI
ADK_DELIVERY_NOTE_INV_DATE	21	Date	8		R	Leverantörsfakturans fakturadatum
ADK_DELIVERY_NOTE_TIMES_TAMP	22	Date	10		R	Tidsstämpel
ADK_DELIVERY_NOTE_LOCAL_REMARK	23	C	120		RW	Kommentar

Inkommande följesedelsrader – ADK_DB_DELIVERY_NOTE_ROW

Program: Visma Administration 2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_OOI_ROW_ARTICLE_NUMBER	0	C	40		RW	Artikelnummer
ADK_OOI_ROW_SUPPLIER_ARTICLE_NUMBER	1	C	40		RW	Leverantörens artikelnummer.
ADK_OOI_ROW_QUANTITY1	2	D	12	4	RW	Antalsfält. Inkommande följesedel: Levererat antal.
ADK_OOI_ROW_QUANTITY2	3	D	12	4	RW	Extra antalsfält. Inkommande följesedel: Restnoterat på kopplad beställningsrad.
ADK_OOI_ROW_QUANTITY3	4	D	12	4	RW	Extra antalsfält. Inkommande följesedel: Väntas på kopplad beställningsrad.
ADK_OOI_ROW_PRICE_EACH_CURRENT_CURRENCY	5	D	12	2	RW	Pris per enhet i aktuell valuta. Inklusive eller exklusive moms.
ADK_OOI_ROW_PRICE2	6	D	12	2	R	Används inte i inkommande följesedel.
ADK_OOI_ROW_DISCOUNT	7	D	10	2	R	Används inte i inkommande följesedel.
ADK_OOI_ROW_AMOUNT_DISCOUNT_FLAG	8	L	1		R	Används inte i inkommande följesedel.
ADK_OOI_ROW_AMOUNT_CURRENT_CURRENCY	9	D	14	2	RW	Totalt belopp i aktuell valuta inklusive eller exklusive moms. Normalt positivt.
ADK_OOI_ROW_AMOUNT_DOMESTIC_CURRENCY	10	D	14	2	R	Totalt belopp i inhemsk valuta utan moms. Normalt positivt.
ADK_OOI_ROW_ACCOUNT_NUMBER	11	C	6		R	Kontonummer.
ADK_OOI_ROW_PROFIT_CENTRE	12	C	6		RW	Resultatenhet. Definieras i eget register.
ADK_OOI_ROW_PROJECT	13	C	10		RW	Projekt. Definieras i eget register.
ADK_OOI_ROW_TEXT	14	C	60		RW	Används för text. Exempelvis artikelbenämning i offert, order, faktura.
ADK_OOI_ROW_UNIT	15	C	4		RW	Enhetskod. Definieras i eget register.
ADK_OOI_ROW_VAT_CODE	16	C	1		R	Används inte i inkommande följesedel.
ADK_OOI_ROW_NOTE_OF_OUTSTANDING_ORDERS	17	L	1		R	Används inte i inkommande följesedel.
ADK_OOI_ROW_DATE2	18	Date	8		R	Extra datumfält. Används i order för beräknat leveransdatum och i faktura för restnoterat leveransdatum.
ADK_OOI_ROW_TYPE_OF_ROW	19	C	1		R	Radtyp. Användning beror på typ, normalt blank. 'T' = Textrad, 'D' – deltotal (på samlingsfaktura rad med ordersumma) 'R' – Råntrader.
ADK_OOI_ROW_CONTRIBUTION_DEGREE	20	D	6	1	R	Täckningsgrad.
ADK_OOI_ROW_CONTRIBUTION_MARGIN	21	D	14	2	R	Täckningsbidrag. Blankas om noll.
ADK_OOI_ROW_PRINT	22	L	1		R	Används inte i inkommande följesedel.
ADK_OOI_ROW_PRINT2	23	L	1		R	Används inte i inkommande följesedel.
	24	U	0		R	Används inte.
ADK_OOI_ROW_BAR_CODE	25	C	32		RW	Streckkod.
ADK_OOI_ROW_DATE3	26	Date	8		R	Används inte i inkommande följesedel.
ADK_OOI_ROW_ARTICLE_TYPE	27	C	1		R	Används inte i inkommande följesedel.
ADK_OOI_ROW_FROM_TYPE	28	C	1		RW	Dokumenttyp
ADK_OOI_ROW_FROM_DOCUMENT	29	D	15	0	RW	Dokumentnummer
ADK_OOI_ROW_FROM_DOCUMENT	30	D	9	0	RW	Dokumentrad
ADK_OOI_ROW_ROWNUMBER	31	D	9	0	R	Radnummer

ADK_OOI_ROW_CONNECTION_TYPE	32	C	3		R	Kopplingstyp
ADK_OOI_ROW_CONNECTION_DOCUMENT	33	D	15	0	R	Kopplingsdokument
ADK_OOI_ROW_CONNECTION_DOCROW	34	D	9	0	R	Kopplingsnummer
ADK_OOI_ROW_SERIAL_NUMBER	35	C	30		RW	Serienummer, ej synligt i programmets gränssnitt
ADK_OOI_ROW_BATCH_NUMBER	36	C	30		RW	Batchnummer, ej synligt i programmets gränssnitt
ADK_OOI_ROW_BEST_BEFORE	37	Date	8		RW	Bäst före-datum, ej synligt i programmets gränssnitt
ADK_OOI_ROW_NOTIFY_QT1	38	D	12	4	R	Aviserat antal
ADK_OOI_ROW_NOTIFY_QT2	39	D	12	4	R	Restaviserat antal
ADK_OOI_ROW_PIN_BOOKING	40	L	1		R	Används inte i inkommande följesedel
ADK_OOI_ROW_NUM_PIN_BOOKING	41	D	9	0	R	Används inte i inkommande följesedel
ADK_OOI_ROW_LABOUR_COST	42	L	1		RW	Används inte i inkommande följesedel
ADK_OOI_ROW_QT_BOOKING	43	D	12	3	R	Kvantitet (används av leverantörsfaktura)
ADK_OOI_ROW_ORDER_BASE_TYPE	44	C	1		R	Används inte i inkommande följesedel
ADK_OOI_ROW_USER_ID	45	C	15		R	Medarbetarid
ADK_OOI_ROW_STRUCTURE_ORDER	46	D	9	0	R	Används inte i inkommande följesedel
ADK_OOI_ROW_PACKAGE	47	L	1		R	Används inte i inkommande följesedel
ADK_OOI_ROW_STRUCTURE	48	L	1		R	Används inte i inkommande följesedel
ADK_OOI_ROW_TIMESTAMP	49	Date	10		R	Tidsstämpel
ADK_OOI_ROW_TAX_REDUCTION_TYPES_ID	50	C	4		R	Används inte i inkommande följesedel.
ADK_OOI_ROW_HOURS_WORKED	51	D	7	1	R	Används inte i inkommande följesedel.
ADK_OOI_ROW_H_SCHAB_PROC	52	D	6	2	R	Används inte i inkommande följesedel.

Kolli – ADK_DB_PACKAGE_HEAD

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_PACKAGE_HEAD_PACKAGE_NUMBER	0	C	20		RW	Kollinummer
ADK_PACKAGE_HEAD_PACKAGE_REFERENCE	1	C	20		RW	Kollireferens
ADK_PACKAGE_HEAD_QUANTITY	2	D	9	0	RW	165nta li kolli
ADK_PACKAGE_HEAD_TYPE	3	C	4		RW	Förpackningstyp
ADK_PACKAGE_HEAD_GROSS_WEIGHT	4	D	12	4	RW	Bruttovikt
ADK_PACKAGE_HEAD_GROSS_VOLUME	5	D	12	4	RW	Bruttovolym
ADK_PACKAGE_HEAD_NET_WEIGHT	6	D	12	4	RW	Nettovikt
ADK_PACKAGE_HEAD_NET_VOLUME	7	D	12	4	RW	Nettovolum
ADK_PACKAGE_HEAD_AREA	8	D	12	4	RW	Flakmeter
ADK_PACKAGE_HEAD_IDENT	9	C	30		RW	Kolliidentitet
ADK_PACKAGE_HEAD_BATCH_NUMBER	10	C	30		RW	Batchnummer
ADK_PACKAGE_HEAD_CONNECTION_DOCUMENT	11	D	15	0	RW	Kopplingsdokumentsnummer
ADK_PACKAGE_HEAD_BEST_BEFORE	12	Date	8		RW	Bäst före datum
ADK_PACKAGE_HEAD_ROWS	13	Data	1		RW	Rader
ADK_PACKAGE_HEAD_NROWS	14	D	9	0	RW	Antal rader

Kollirader – ADK_DB_PACKAGE_ROW

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_PACKAGE_ROW_ROWNR	0	D	9	0	R	Kolliradnummer
ADK_PACKAGE_ROW_NAME	1	C	60		R	Artikelbenämning
ADK_PACKAGE_ROW_QUANTITY	2	D	12	4	RW	Antal
ADK_PACKAGE_ROW_ARTICLE_NUMBER	3	C	40		R	Artikelnummer
ADK_PACKAGE_ROW_SERIAL_NUMBER	4	C	30		RW	Serienummer
ADK_PACKAGE_ROW_BATCH_NUMBER	5	C	30		RW	Batchnummer
ADK_PACKAGE_ROW_BEST_BEFORE	6	Date	8		RW	Bäst före datum
ADK_PACKAGE_ROW_ORDER_ROW_NUMBER	7	D	9	0	RW	Orderradsnummer

Leveransaviseringskolli – ADK_DB_IMP_PACKAGE_HEAD

Program: Visma Administration 2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_IMP_PACKAGE_HEAD_PACKAGE_NUMBER	0	C	20		R	Kollinummer
ADK_IMP_PACKAGE_HEAD_QUANTITY	1	D	9	0	R	166nta li kolli
ADK_IMP_PACKAGE_HEAD_TYPE	2	C	4		R	Förpackningstyp
ADK_IMP_PACKAGE_HEAD_GROSS_WEIGHT	3	D	12	4	R	Bruttovikt
ADK_IMP_PACKAGE_HEAD_GROSS_VOLUME	4	D	12	4	R	Bruttovolym
ADK_IMP_PACKAGE_HEAD_NET_WEIGHT	5	D	12	4	R	Nettovikt
ADK_IMP_PACKAGE_HEAD_NET_VOLUME	6	D	12	4	R	Nettovolum
ADK_IMP_PACKAGE_HEAD_AREA	7	D	12	4	R	Flakmeter
ADK_IMP_PACKAGE_HEAD_IDENTIFIER	8	C	30		R	Kolliidentitet
ADK_IMP_PACKAGE_HEAD_BATCH_NUMBER	9	C	30		R	Batchnummer
ADK_IMP_PACKAGE_HEAD_CONNECTION_DOCUMENT	10	D	15	0	R	Kopplingsdokumentsnummer
ADK_IMP_PACKAGE_HEAD_BEST_BEFORE	11	Date	8		R	Bäst före datum
ADK_IMP_PACKAGE_HEAD_ROWS	12	Data	1		R	Rader
ADK_IMP_PACKAGE_HEAD_ROWS	13	D	9	0	R	Antal rader

Leveransaviseringskollirader – ADK_DB_IMP_PACKAGE_ROW

Program: Visma Administration 2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_IMP_PACKAGE_ROW_ROWNR	0	D	9	0	R	Kolliradnummer

ADK_IMP_PACKAGE_ROW_NAME	1	C	60		R	Artikelbenämning
ADK_IMP_PACKAGE_ROW_QUANTITY	2	D	12	4	R	Antal
ADK_IMP_PACKAGE_ROW_ARTICLE_NUMBER	3	C	40		R	Artikelnummer
ADK_IMP_PACKAGE_ROW_SERIAL_NUMBER	4	C	30		R	Serienummer
ADK_IMP_PACKAGE_ROW_BATCH_NUMBER	5	C	30		R	Batchnummer
ADK_IMP_PACKAGE_ROW_BEST_BEFORE	6	Date	8		R	Bäst före datum

Inventering – ADK_DB_INVENTORY_ARTICLE

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_INVENTORY_ARTICLE_ARTICLE_NUMBER	0	C	40		RW	Artikelnummer, unikt
ADK_INVENTORY_ARTICLE_ARTICLE_NAME	1	C	60		R	Artikelns benämning på det egna språket. Benämning på andra språk anges i annat reg.
ADK_INVENTORY_ARTICLE_ARTICLE_GROUP	2	C	6		R	Artikelgrupp. Artikelgrupper definieras i eget register.
ADK_INVENTORY_ARTICLE_ARTICLE_PLACE_IN_STOCK	3	C	6		R	Lagerplats.
ADK_INVENTORY_ARTICLE_ARTICLE_UNIT_CODE	4	C	4		R	Enhetskod. Enhetskoder definieras i eget register.
ADK_INVENTORY_ARTICLE_QUANTITY	5	D	14**	4**	RW	Inventerat antal som ännu inte satts som definitivt (under pågående inventering). Detta antal flyttas över till ADK_ARTICLE_INVENTORY_QUANTITY när inventeringsjournalen körs.
ADK_INVENTORY_ARTICLE_DATE	6	Date	8		RW	Inventeringsdatum under pågående inventering.
ADK_INVENTORY_ARTICLE_QUANTITY_IN_STOCK	7	D	14	4	R	Aktuellt lagerantal vid inventeringstillfället.
ADK_INVENTORY_ARTICLE_DIFFERENCE	8	C	14		R	Differens mellan lagerantal och inventerat antal.
ADK_INVENTORY_ARTICLE_LAST_DATE	9	Date	8		R	Datum för senaste inventering. Det datum som sätts upp som inventeringsdatum vid inmatning av inventering.
ADK_INVENTORY_ARTICLE_LAST_DIFFERENCE	10	D	14	4	R	Differens vid senaste inventering.
ADK_INVENTORY_ARTICLE_ACCUMULATED_DIFFERENCE	11	D	14	4	R	Differens, ackumulerat. Uppdateras vid inventeringsjournalen.
ADK_INVENTORY_ARTICLE_FLAG	12	L	1		RW	Sann betyder att ej journalförd inventerat antal finns i fältet ADK_INVENTORY_ARTICLE_QUANTITY. Ändrar man detta fält vid AdkUpdate eller AdkAdd så nollas inventeringsraden bort.
ADK_INVENTORY_ARTICLE_BARCODE	13	C	32		R	Streckkod

** Antalet tillåtna tecken bestäms av enheten på artikeln. Det reserveras alltid 4 tecken för decimaler och 1 tecken för decimaltecken om det är fler än 0 decimaler. Är det en artikel med 0 decimaler går det alltså att skriva 8 tecken, är det en artikel med fler än 0 decimaler går det att skriva 7 tecken innan decimaltecknet och antalet decimaler efter.

Manuell inleverans – ADK_DB_MANUAL_DELIVERY_IN

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_MANUAL_DELIVERY_IN_DOCUMENT_NUMBER	0	D	15	0	RW	Dokumentnummer som raden tillhör, manuell inleverans.

ADK_MANUAL_DELIVERY_IN_DATE	1	Date	8		RW	Transaktionsdatum.
ADK_MANUAL_DELIVERY_IN_ARTICLE_NUMBER	2	C	40		RW	Artikelnummer.
ADK_MANUAL_DELIVERY_IN_ARTICLE_NAME	3	C	60		R	Artikels benämning på det egna språket. Benämning på andra språk anges i annat fält, se nedan.
ADK_MANUAL_DELIVERY_IN_UNIT	4	C	4		R	Enhetskod. Enhetskoderna definieras i enhetsregister.
ADK_MANUAL_DELIVERY_IN_QUANTITY	5	D	12	4	RW	Antal.
ADK_MANUAL_DELIVERY_IN_PRICE_EACH	6	D	12	4	RW	Pris
ADK_MANUAL_DELIVERY_IN_CARGO_AMOUNT_EACH	7	D	10	4	RW	Fraktkostnad per styck.
ADK_MANUAL_DELIVERY_IN_TEXT	8	C	60		RW	Kommentar.
ADK_MANUAL_DELIVERY_IN_ACCOUNT	9	C	6		RW	Konto.
ADK_MANUAL_DELIVERY_IN_PROFIT_CENTRE	10	C	6		RW	Kod för resultatenhet. Resultatenheter definieras i ett eget register.
ADK_MANUAL_DELIVERY_IN_PROJECT	11	C	10		RW	Projekt. Definieras i eget register.
ADK_MANUAL_DELIVERY_IN_CANCELLED	12	L	1		RW	Sann indikerar att dokumentet är makulerat.
ADK_MANUAL_DELIVERY_IN_JOURNAL_PRINTED	13	L	1		R	Sann indikerar att dokumentet är journalfört.
ADK_MANUAL_DELIVERY_IN_TIMESTAMP	14	Date	10		R	Tidsstämpel
ADK_MANUAL_DELIVERY_IN_SERIAL_NUMBER	15	C	30		RW	Serienummer, ej synligt i programmets gränssnitt
ADK_MANUAL_DELIVERY_IN_BATCH_NUMBER	16	C	30		RW	Batchnummer, ej synligt i programmets gränssnitt
ADK_MANUAL_DELIVERY_IN_BEST_BEFORE	17	Date	8		RW	Bäst före-datum, ej synligt i programmets gränssnitt

Manuell utleverans – ADK_DB_MANUAL_DELIVERY_OUT

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_MANUAL_DELIVERY_OUT_DOCUMENT_NUMBER	0	D	15	0	RW	Dokumentnummer som raden tillhör, manuell utleverans.
ADK_MANUAL_DELIVERY_OUT_DATE	1	Date	8		RW	Transaktionsdatum.
ADK_MANUAL_DELIVERY_OUT_ARTICLE_NUMBER	2	C	40		RW	Artikelnummer.
ADK_MANUAL_DELIVERY_OUT_ARTICLE_NAME	3	C	60		R	Artikels benämning på det egna språket. Benämning på andra språk anges i annat fält, se nedan.
ADK_MANUAL_DELIVERY_OUT_UNIT	4	C	4		R	Enhetskod. Enhetskoderna definieras i enhetsregister.
ADK_MANUAL_DELIVERY_OUT_QUANTITY	5	D	12	4	RW	Antal.
ADK_MANUAL_DELIVERY_OUT_TEXT	6	C	60		RW	Kommentar.
ADK_MANUAL_DELIVERY_OUT_ACCOUNT	7	C	6		RW	Konto.
ADK_MANUAL_DELIVERY_OUT_PROFIT_CENTRE	8	C	6		RW	Kod för resultatenhet. Resultatenheter definieras i ett eget register.
ADK_MANUAL_DELIVERY_OUT_PROJECT	9	C	10		RW	Projekt. Definieras i eget register.
ADK_MANUAL_DELIVERY_OUT_CANCELLED	10	L	1		RW	Sann indikerar att dokumentet är makulerat.
ADK_MANUAL_DELIVERY_OUT_JOURNAL_PRINTED	11	L	1		R	Sann indikerar att dokumentet är journalfört.
ADK_MANUAL_DELIVERY_OUT_TIMESTAMP	12	Date	10		R	Tidsstämpel
ADK_MANUAL_DELIVERY_OUT_SERIAL_NUMBER	13	C	30		RW	Serienummer, ej synligt i programmets gränssnitt

ADK_MANUAL_DELIVERY_OUT_BATCH_NUMBER	14	C	30		RW	Batchnummer, ej synligt i programmets gränssnitt
ADK_MANUAL_DELIVERY_OUT_BEST_BEFORE	15	Date	8		RW	Bäst före-datum, ej synligt i programmets gränssnitt

Projekt – ADK_DB_PROJECT

Program: Visma Administration 1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_PROJECT_CODE_OF_PROJECT	0	C	10		RW	Kod för projekt, unikt.
ADK_PROJECT_NAME	1	C	50		RW	Projektets benämning.
ADK_PROJECT_DATE_OF BEGINNING	2	Date	8		RW	Projektets början.
ADK_PROJECT_DATE_OF END	3	Date	8		RW	Projektets beräknade slut.
ADK_PROJECT_DATE_OF CLOSURE	4	Date	8		RW	Datum då projektet avslutas.
ADK_PROJECT_RESPONSIBLE_OF_PROJECT	5	C	20		RW	Namn på ansvarig.
ADK_PROJECT_CUSTOMER_NUMBER (2)	6	C	16		RW	Kundnummer.
ADK_PROJECT_CUSTOMER_NAME (2)	7	C	50		RW	Kundens namn.
ADK_PROJECT_OUR_ORDER_NUMBER (2)	8	D	15	0	RW	Vårt ordernummer.
ADK_PROJECT_CUSTOMER_ORDER_NUMBER (2)	9	C	25		RW	Kundens ordernummer.
ADK_PROJECT_CUSTOMER_REFERENCE_NAME	10	C	20		RW	Kundens kontaktperson. (Förening: Projektets kontaktperson)
ADK_PROJECT_TELEPHONE	11	C	20		RW	Telefon.
ADK_PROJECT_FAX	12	C	20		RW	Fax
ADK_PROJECT_NOTE ABOUT WORKPLACE	13	C	35		RW	Anteckningsfält för uppgifter om arbetsplats. (Förening: Arrangemang)
ADK_PROJECT_NOTE ABOUT WORKPLACE2	14	C	35		RW	Anteckningsfält för uppgifter om arbetsplats. (Förening: Arrangemang rad 2)
ADK_PROJECT_NOTE1	15	C	60		RW	Anteckningsfält för fri användning.
ADK_PROJECT_NOTE2	16	C	60		RW	Anteckningsfält för fri användning.
ADK_PROJECT_NOTE3	17	C	60		RW	Anteckningsfält för fri användning.
ADK_PROJECT_NOTE4	18	C	60		RW	Anteckningsfält för fri användning.
ADK_PROJECT_CHART_OF ACCOUNTS_TYPE	19	C	8		R	Kontoplanstyp
ADK_PROJECT_CLOSED	20	L	1		RW	Avslutat projekt
ADK_PROJECT_TIMESTAMP	21	Date	10		R	Tidsstämpel
ADK_PROJECT_DOCUMENT_PATH	22	C	160		RW	Dokument (sökväg)

(2) Fältet används endast i Visma Administration 1000/2000.

Resultatenheter – ADK_DB_CODE_OF_PROFIT_CENTRE

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_PROFIT_CENTRE_YEAR_ID	0	C	1		R	Årsid, 'A' – 'Z'
ADK_CODE_OF_PROFIT_CENTRE_CODE	1	C	6		RW	Kod resultatenhet.
ADK_CODE_OF_PROFIT_CENTRE_NAME	2	C	20		RW	Resultatenhetens namn.
ADK_CODE_OF_PROFIT_CENTRE_TIMESTAMP	3	Date	10		R	Tidsstämpel

Företagsinställningar/föreningsinställningar – ADK_DB_PRM

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_PRM_NAME	0	C	50		RW	Företagets/föreningens namn.
ADK_PRM_MAILING_ADDRES S	1	C	35		RW	Postadress, gata, box.
ADK_PRM_VISITING_ADDRES S	2	C	35		RW	Besöksadress.
ADK_PRM_ZIPCODE	3	C	12		RW	Även internationella postnr.
ADK_PRM_CITY	4	C	24		RW	Ortsnamn.
ADK_PRM_COUNTRY	5	C	24		RW	Landskod.
ADK_PRM_TELEPHONE	6	C	20		RW	Telefonnummer.
ADK_PRM_TELEPHONE2	7	C	20		RW	Alternativt telefonnummer.
ADK_PRM_FAX	8	C	20		RW	Faxnummer.
ADK_PRM_BGIRO	9	C	16		R	Bankgironummer.
ADK_PRM_PGIRO	10	C	13		R	Plusgironummer.
ADK_PRM_ORGANISATION_N UMBER	11	C	14		R	Organisationnummer.
ADK_PRM_VAT_REGISTER_N UMBER	12	C	14		RW	Momsregistreringsnummer.
ADK_PRM_VAT_NUMBER	13	C	14		RW	VAT-nr. Numera samma fält som momsregistreringsnummer och pekas om mot detta fält.
ADK_PRM_CODE_OF_LANGU AGE (3)	14	C	4		RW	Inhemska språkkoden.
ADK_PRM_CODE_OF_CURRE NCY (3)	15	C	4		RW	Inhemska valutan.
ADK_PRM_CODE_OF_BRANC H (3)	16	C	6		R	Branchkod.
ADK_PRM_EMAIL	17	C	70		RW	Företagets/föreningens e-postadress
ADK_PRM_WWW	18	C	160		RW	Företagets/föreningens www-adress.
ADK_PRM_SEAT	19	C	24		R	Företagets/föreningens säte.
ADK_PRM_DELIVERY_NAME (1)	20	C	50		R	Namn för leverans.
ADK_PRM_DELIVERY_ADDRE SS (1)	21	C	35		R	Adress för leverans.
ADK_PRM_DELIVERY_ADDRE SS2 (1)	22	C	35		R	Adress2 för leverans.
ADK_PRM_DELIVERY_ZIPCO DE (1)	23	C	12		R	Postnummer för leverans.
ADK_PRM_DELIVERY_CITY (1)	24	C	24		R	Ort för leverans.
ADK_PRM_DELIVERY_MAILIN G_ADDRESS (1)	25	C	37		R	Postadress för leverans.
ADK_PRM_DELIVERY_COUNT RY (1)	26	C	24		R	Land för leverans.
ADK_PRM_DELIVERY_TELEP HONE (1)	27	C	20		R	Telefonnummer för leveransadress.
ADK_PRM_DELIVERY_REFER ENCE (1)	28	C	25		R	Referens för leveransadress.
ADK_PRM_EANLOC	29	C	20		R	EAN-lokaliseringsnummer.
ADK_PRM_DELIVERY_EANLO C (1)	30	C	20		R	EAN-lokaliseringsnummer för leveransadress.
ADK_PRM_IBAN_NUMBER (3)	31	C	46		R	IBAN-nummer.
ADK_PRM_SWIFT_ADRESS (3)	32	C	11		R	BIC.
ADK_PRM_BANK_NAME (3)	33	C	50		R	Bankens namn.
ADK_PRM_BANK_ADRESS (3)	34	C	35		R	Bankens adress.
ADK_PRM_BANK_ADRESS2 (3)	35	C	35		R	Bankens adress2.
ADK_PRM_BANK_ZIPCODE (3)	36	C	12		R	Bankens postnummer.
ADK_PRM_BANK_CITY (3)	37	C	25		R	Bankens ortsnamn.
ADK_PRM_CODE_OF_COUNT RY (3)	38	C	4		R	Bankens land.
ADK_PRM_ACCOUNT_NUMBE R (3)	39	C	30		R	Kontonummer.
ADK_PRM_ADDRESS_CODE_ OF_COUNTRY (3)	40	C	4		RW	Landskod
ADK_PRM_DELIVERY_CODE_ OF_COUNTRY (1)	41	C	4		R	Landskod leveransadress

ADK_PRM_TAX_REDUCTION_ACTIVATED (3)	42	L	1		R	Företaget fakturerar husarbete
ADK_PRM_TAX_REDUCTION_PERCENT (3)	43	D	4	0	R	Skattereduktion i % (RUT)
ADK_PRM_TAX_REDUCTION_MAX_AMOUNT (3)	44	D	9	0	R	Maxbelopp per person (RUT)
ADK_PRM_PROGRAM	45	C	8		R	Program
ADK_PRM_DATABASE_VERSION	46	D	5	2	R	Databasversion
ADK_PRM_QUANTITY_ACCOUNTING_ACTIVATED	47	L	1		R	Kvantitetsredovisning (aktiverad)
ADK_PRM_TRANSACTIONINFO_ACTIVATED	48	L	1		R	Transaktionsinfo (aktiverat)
ADK_PRM_SUBACCOUNTS_ACTIVATED (1)	49	L	1		R	Subkonton (aktiverat)
ADK_PRM_LANGUAGE_CODE (3)	50	C	4		R	Språkkod (företagets)
ADK_PRM_CURRENCY_CODE (3)	51	C	4		R	Valutakod (företagets)
ADK_PRM_DEFAULT_SUPPLIER_INVOICE_ATTEST	52	C	20		R	Attest leverantörsfakturor (normalvärde)
ADK_PRM_DEFAULT_SUPPLIER_PAYMENT_TERMS	53	C	8		R	Betalningsvillkor, leverantör (normalvärde)
ADK_PRM_DEFAULT_SUPPLIER_DELIVERY_TERMS	54	C	8		R	Leveransvillkor, leverantör (normalvärde)
ADK_PRM_DEFAULT_SUPPLIER_DELIVERY_WAY	55	C	8		R	Leveranssätt, leverantör (normalvärde)
ADK_PRM_DEFAULT_SUPPLIER_REFERENCE	56	C	25		R	Vår referens, leverantör (normalvärde)
ADK_PRM_DEFAULT_SUPPLIER_INVOICE_ATTEST_ACTIVATED	57	L	1		R	Attest leverantörsfakturor (aktiverat)
ADK_PRM_DEFAULT_PRICE_LIST_ROUND_FROM1	58	D	12	2	R	Avrunda från 1 (normalvärde)
ADK_PRM_DEFAULT_PRICE_LIST_ROUND_FROM2	59	D	12	2	R	Avrunda från 2 (normalvärde)
ADK_PRM_DEFAULT_PRICE_LIST_ROUND_FROM3	60	D	12	2	R	Avrunda från 3 (normalvärde)
ADK_PRM_DEFAULT_PRICE_LIST_ROUND_FROM4	61	D	12	2	R	Avrunda från 4 (normalvärde)
ADK_PRM_DEFAULT_PRICE_LIST_ROUND1	62	D	12	2	R	Avrundning 1 (normalvärde)
ADK_PRM_DEFAULT_PRICE_LIST_ROUND2	63	D	12	2	R	Avrundning 2 (normalvärde)
ADK_PRM_DEFAULT_PRICE_LIST_ROUND3	64	D	12	2	R	Avrundning 3 (normalvärde)
ADK_PRM_DEFAULT_PRICE_LIST_ROUND4	65	D	12	2	R	Avrundning 4 (normalvärde)
ADK_PRM_DEFAULT_ARTICLE_GROUP	66	C	6		R	Artikelgrupp (normalvärde)
ADK_PRM_DEFAULT_ARTICLE_ACCOUNTING_CODE	67	C	6		R	Artikelkontering (normalvärde)
ADK_PRM_DEFAULT_ARTICLE_UNIT_CODE	68	C	4		R	Enhetskod (normalvärde)
ADK_PRM_DEFAULT_ARTICLE_STOCK_GOODS (2)	69	L	1		R	Lagervara (normalvärde)
ADK_PRM_SNI_CODE	70	C	6		R	SNI-kod
ADK_PRM_VERIFICATION_SERIES_ACTIVATED	71	L	1		R	Verifikationsserier (aktiverat)
ADK_PRM_PRICE_4_DECIMALS_ACTIVATED	72	L	1		R	Pris med 4 decimaler (aktiverat)
ADK_PRM_DEFAULT_ARTICLE_BARCODE_TYPE	73	D	4	0	R	Streckkodstyp (normalvärde)
ADK_PRM_BARCODE_ACTIVATED	74	L	1		R	Streckkodstyp (aktiverat)
ADK_PRM_AKI_BG_BETSATT	75	C	8		R	Betalsätt bankgiro i återrapporteringar av inbetalningar (normalvärde).
ADK_PRM_AKI_PG_BETSATT	76	C	8		R	Betalsätt plusgiro i återrapporteringar av inbetalningar (normalvärde).
ADK_PRM_TAX_REDUCTION_PERCENT_BUILDING_WORK (3)	77	D	4	0	R	Skattereduktion i % (ROT)

ADK_PRM_TAX_REDUCTION_MAX_AMOUNT_BUILDING_WORK (3)	78	D	9	0	R	Maxbelopp per person (ROT)
ADK_PRM_TAX_REDUCTION_DEFAULT_TYPE (3)	79	D	4	0	R	Defaultinställning för RUT/ROT på nya avtal/order/fakturer (0=RUT, 1=ROT)
ADK_PRM_AKI_OCR	80	L	1		R	OCR-nr används i kundfaktura och återrapportering av inbetalningar
ADK_PRM_TAX_REDUCTION_MAX_AMOUNT_GT	81	D	9	0	R	Maxbelopp per person (Grön teknik)

(1) Fältet används endast i Visma Administration 2000.

(2) Fältet används endast i Visma Administration 1000/2000.

(3) Fältet används endast i Visma Administration 500/1000/2000.

Artikelbenämning på annat språk – ADK_DB_CODE_OF_ARTICLE_NAME

Program: Visma Administration 500/1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_ARTICLE_NAME_LANGUAGE_CODE	0	C	4		RW	Kod för språket.
ADK_CODE_OF_ARTICLE_NAME_ARTICLE_NUMBER	1	C	40		RW	Artikelnr.
ADK_CODE_OF_ARTICLE_NAME_NAME	2	C	60		RW	Benämning.
ADK_CODE_OF_ARTICLE_NAME_NAME2	3	C	60		RW	Extra benämningstext.
ADK_CODE_OF_ARTICLE_NAME_TIMESTAMP	4	Date	10		R	Tidsstämpel

Artikelgrupper – ADK_DB_CODE_OF_ARTICLE_GROUP

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_ARTICLE_GROUP_CODE	0	C	6		RW	Kod för artikelgruppen.
ADK_CODE_OF_ARTICLE_GROUP_TEXT	1	C	25		RW	Benämning.
ADK_CODE_OF_ARTICLE_GROUP_TIMESTAMP	2	Date	10		R	Tidsstämpel

Artikelontering – ADK_DB_CODE_OF_ARTICLE_ACCOUNT

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_ARTICLE_ACCOUNT_CODE	0	C	6		RW	Konteringskod.
ADK_CODE_OF_ARTICLE_ACCOUNT_TEXT	1	C	25		RW	Benämning.
ADK_CODE_OF_ARTICLE_ACCOUNT_SALES	2	C	6		RW	Konto för försäljning.
ADK_CODE_OF_ARTICLE_ACCOUNT_EXPORT_SALES	3	C	6		RW	Konto för exportförsäljning.
ADK_CODE_OF_ARTICLE_ACCOUNT_SALES_EU	4	C	6		RW	Konto för EU-försäljning.
ADK_CODE_OF_ARTICLE_ACCOUNT_PURCHASE (2)	5	C	6		RW	Konto för inköp.
ADK_CODE_OF_ARTICLE_ACCOUNT_STOCK (2)	6	C	6		RW	Konto för lager (tillgångskonto).
ADK_CODE_OF_ARTICLE_ACCOUNT_STOCK_CHANGE (2)	7	C	6		RW	Konto för lagerförändring.
ADK_CODE_OF_ARTICLE_ACCOUNT_PROFIT_CENTRE	8	C	6		RW	Resultatenhet.
ADK_CODE_OF_ARTICLE_	9	C	1		RW	Momskod.

ACCOUNT_VAT						
ADK_CODE_OF_ARTICLE_ACCOUNT_VAT_EU	10	C	6		RW	Konto moms, EU-försäljning
ADK_CODE_OF_ARTICLE_ACCOUNT_SERVICE	11	L	1		RW	Tjänst
ADK_CODE_OF_ARTICLE_ACCOUNT_BUILD (3)	12	C	6		RW	Kontonummer vid byggtjänst
ADK_CODE_OF_ARTICLE_ACCOUNT_MIDMAN	13	C	6		RW	Kontonummer vid mellanman
ADK_CODE_OF_ARTICLE_ACCOUNT_REDUCED_GPF	14	C	6		R	Används inte från version 4.5
ADK_CODE_OF_ARTICLE_ACCOUNT_PROJECT	15	C	10		RW	Projekt
ADK_CODE_OF_ARTICLE_ACCOUNT_TIMESTAMP	16	Date	10		R	Tidsstämpel

(2) Fältet används endast i Visma Administration 1000/2000.

(3) Fältet används endast i Visma Administration 500/1000/2000.

Betalningsvillkor – ADK_DB_CODE_OF_TERMS_OF_PAYMENT

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_TERMS_OF_PAYMENT_CODE	0	C	8		RW	Kod för betalningsvillkor, unikt.
ADK_CODE_OF_TERMS_OF_PAYMENT_TEXT	1	C	30		RW	Text för betalningsvillkor.
ADK_CODE_OF_TERMS_OF_PAYMENT_PAYMENT_CODE	2	C	8		RW	Koder som anger hur betalvillkor och förfalldatum ska behandlas av programmet, ska vara någon av följande: numeriskt = föreslaget förfalldatum är fakturadatum + innehållet i fältet, P= postförskott, K= kontant.
ADK_CODE_OF_TERMS_OF_PAYMENT_TIMESTAMP	3	Date	10		R	Tidsstämpel

Betalsätt – ADK_DB_CODE_OF_WAY_OF_PAYMENT

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_WAY_OF_PAYMENT_CODE	0	C	8		RW	Kod för betalsätt, unikt.
ADK_CODE_OF_WAY_OF_PAYMENT_TEXT	1	C	20		RW	Text för betalsätt.
ADK_CODE_OF_WAY_OF_PAYMENT_ACCOUNT	2	C	6		RW	Kontonummer.
ADK_CODE_OF_WAY_OF_PAYMENT_TIMESTAMP	3	Date	10		R	Tidsstämpel

Distrikt – ADK_DB_CODE_OF_DISTRICT

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_DISTRICT_CODE	0	C	8		RW	Kod för distrikt, unikt.
ADK_CODE_OF_DISTRICT_TEXT	1	C	25		RW	Benämning på distrikt.
ADK_CODE_OF_DISTRICT_TIMESTAMP	2	Date	10		R	Tidsstämpel

Enhetskoder –ADK_DB_CODE_OF_UNIT

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_UNIT_CODE	0	C	4		RW	Enhetskod.
ADK_CODE_OF_UNIT_DOMESTIC	1	L	1		R	Sann indikerar att det är den inhemska beteckningen på enheten.
ADK_CODE_OF_UNIT_LANGUAGE	2	C	4		RW	Språkkod. Används inte i den post som har fältet "DOMESTIC" sann.
ADK_CODE_OF_UNIT_TEXT	3	C	12		RW	Texten som representerar enheten.
ADK_CODE_OF_UNIT_DECIMALS	4	D	4	0	RW	Antal decimaler.
ADK_CODE_OF_UNIT_ISO	5	C	4		RW	ISO
ADK_CODE_OF_UNIT_TIMESTAMP	6	Date	10		R	Tidsstämpel

Fria kategorier 1..10 – ADK_DB_FREE_CATEGORY_1..10

Program: Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_FREE_CATEGORY_CATEGORY	0	C	4		RW	Kategorins Beteckning
ADK_FREE_CATEGORY_NAME	1	C	25		RW	Kategorins Benämning
ADK_FREE_CATEGORY_TIMESTAMP	2	Date	10		R	Tidsstämpel

Kundkategorier – ADK_DB_CODE_OF_CUSTOMER_CATEGORY

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_CUSTOMER_CATEGORY_CODE	0	C	4		RW	Kod för kundkategori, unikt.
ADK_CODE_OF_CUSTOMER_CATEGORY_TEXT	1	C	25		RW	Benämning på kundkategori.
ADK_CODE_OF_CUSTOMER_CATEGORY_MEMBER_CATEGORY (4)	2	L	1		RW	Medlemskategori
ADK_CODE_OF_CUSTOMER_CATEGORY_TIMESTAMP	3	Date	10		R	Tidsstämpel

(4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

Leveranssätt – ADK_DB_CODE_OF_WAY_OF_DELIVERY

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_WAY_OF_DELIVERY_CODE	0	C	8		RW	Kod för leveranssätt, unikt.
ADK_CODE_OF_WAY_OF_DELIVERY_TEXT	1	C	32		RW	Text för leveranssätt.
ADK_CODE_OF_WAY_OF_DELIVERY_TIMESTAMP	2	Date	10		R	Tidsstämpel

Leveransvillkor – ADK_DB_CODE_OF_TERMS_OF_DELIVERY

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
----------	----	-----	-------	-----	------	-------------

ADK_CODE_OF_TERMS_OF_DELIVERY_CODE	0	C	8		RW	Kod för leveransvillkoret, unikt.
ADK_CODE_OF_TERMS_OF_DELIVERY_TEXT	1	C	32		RW	Text för leveransvillkor.
ADK_CODE_OF_TERMS_OF_DELIVERY_TIMESTAMP	2	Date	10		R	Tidsstämpel

Speditörer – ADK_DB_DISPATCHER

Program: Visma Administration 500/1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_DISPATCHER_ID	0	C	7		RW	Kod för speditör, unikt.
ADK_DISPATCHER_TEXT	1	C	30		RW	Benämning på speditör.
ADK_DISPATCHER_OUR_CUSTOMER_NUMBER	2	C	20		RW	Vårt kundnummer hos speditören.
ADK_DISPATCHER_TIMESTAMP	3	Date	10		R	Tidsstämpel

Språkkoder – ADK_DB_CODE_OF_LANGUAGE

Program: Visma Administration 500/1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_LANGUAGE_CODE	0	C	4		RW	Kod för språket.
ADK_CODE_OF_LANGUAGE_TEXT	1	C	20		RW	Språkets benämning.
ADK_CODE_OF_LANGUAGE_ISO	2	L	1		R	ISO
ADK_CODE_OF_LANGUAGE_TIMESTAMP	3	Date	10		R	Tidsstämpel

Säljare – ADK_DB_CODE_OF_SELLER

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_SELLER_SIGNATURE	0	C	10		RW	Säljarens kod/signatur, unikt.
ADK_CODE_OF_SELLER_NAME	1	C	25		RW	Säljarens namn.
ADK_CODE_OF_SELLER_SELLER	2	L	1		RW	Om det är en säljare
ADK_CODE_OF_SELLER_REFERENCE	3	L	1		RW	Om det är en referens
ADK_CODE_OF_SELLER_ATT	4	L	1		RW	Om attest finns
ADK_CODE_OF_SELLER_TIMESTAMP	5	Date	10		R	Tidsstämpel

Valutakoder/valutakurser – ADK_DB_CODE_OF_CURRENCY

Program: Visma Administration 500/1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_CURRENCY_CODE	0	C	4		RW	Kod för valuta, unikt.
ADK_CODE_OF_CURRENCY_COUNTRY	1	C	20		RW	Land i klartext.
ADK_CODE_OF_CURRENCY_UNIT	2	D	9	0	RW	Ex Italienska lire: En lire värd 0,0049 kr. Om kursen är 0,49 blir enheten 100. D v s 100 lire motsvarar 0,49 kr.
ADK_CODE_OF_CURRENCY_DATE_OF_REGISTRATION	3	Date	8		R	Datum för kursregistrering.

ADK_CODE_OF_CURRENCY_SELL	4	D	12	6	RW	Bankens köpkurs.
ADK_CODE_OF_CURRENCY_BUY	5	D	12	6	RW	Bankens säljkurs.
ADK_CODE_OF_CURRENCY_TIMESTAMP	6	Date	10		R	Tidsstämpel

Avvikande leveransadress – ADK_DB_DELIVERY_ADDRESS

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_DELIVERY_CUSTOMER_NUMBER	0	C	16		RW	Kundnummer
ADK_DELIVERY_NAME	1	C	50		RW	Leveransnamn
ADK_DELIVERY_ADDRESS	2	C	35		RW	Adress1
ADK_DELIVERY_ADDRESS2	3	C	35		RW	Adress2
ADK_DELIVERY_EANLOC	4	C	20		RW	EAN
ADK_DELIVERY_VISITING_ADDRESS	5	C	35		RW	Besökadress
ADK_DELIVERY_ZIPCODE	6	C	12		RW	Postnummer
ADK_DELIVERY_CITY	7	C	24		RW	Ort
ADK_DELIVERY_CODE_OF_COUNTRY	8	C	4		RW	Landskod
ADK_DELIVERY_COUNTRY	9	C	24		RW	Land
ADK_DELIVERY_TELEPHONE1	10	C	20		RW	Telefon1
ADK_DELIVERY_TELEPHONE2	11	C	20		RW	Telefon2
ADK_DELIVERY_TELEPHONE3	12	C	20		RW	Telefon3
ADK_DELIVERY_FAX	13	C	20		RW	Fax
ADK_DELIVERY_TIMESTAMP	14	Date	10		R	Tidsstämpel
ADK_DELIVERY_ID	15	D	9	0	R	Id

Er referens, kunder – ADK_DB_CODE_OF_YOUR_REF_CUSTOMER

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_YOUR_REF_CUSTOMER_NUMBER	0	C	16		RW	Kundnummer
ADK_CODE_OF_YOUR_REF_SIGN	1	C	8		RW	Signatur
ADK_CODE_OF_YOUR_REF_NAME	2	C	50		RW	Namn
ADK_CODE_OF_YOUR_REF_TIMESTAMP	3	Date	10		R	Tidsstämpel

Er referens, leverantörer – ADK_DB_CODE_OF_YOUR_REF_SUPPLIER

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_YOUR_REF_SUPPLIER_NUMBER	0	C	16		RW	Leverantörsnummer
ADK_CODE_OF_YOUR_REF_SUPPLIER_SIGN	1	C	8		RW	Signatur
ADK_CODE_OF_YOUR_REF_SUPPLIER_NAME	2	C	50		RW	Namn
ADK_CODE_OF_YOUR_REF_SUPPLIER_TIMESTAMP	3	Date	10		R	Tidsstämpel

Kontakter – ADK_DB_CONTACT

Program: Visma Administration 500/1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CONTACTS_ID	0	D	6		RW	KontaktID
ADK_CONTACTS_SIGN	1	C	8		RW	Signatur
ADK_CONTACTS_NAME	2	C	50		RW	Namn
ADK_CONTACTS_CUSTOME R_NUMBER	3	C	16		RW	Kundnummer
ADK_CONTACTS_SUPPLIER_ NUMBER	4	C	16		RW	Leverantörsnummer
ADK_CONTACTS_TITLE	5	D	9	0	RW	Titel
ADK_CONTACTS_TELEPHON E	6	C	20		RW	Telefon
ADK_CONTACTS_TELEPHON E2	7	C	20		RW	Telefon 2
ADK_CONTACTS_MOBILEPH ONE	8	C	20		RW	Mobiltelefon
ADK_CONTACTS_EMAIL	9	C	70		RW	Epost
ADK_CONTACTS_COMPANY_ NAME	10	C	50		RW	Företagsnamn
ADK_CONTACTS_MAILING_A DDRESS	11	C	35		RW	Postadress
ADK_CONTACTS_MAILING_A DDRESS2	12	C	35		RW	Postadress 2
ADK_CONTACTS_VISITING_A DDRESS	13	C	35		RW	Besöksadress
ADK_CONTACTS_ZIP_CODE	14	C	12		RW	Postnummer
ADK_CONTACTS_CITY	15	C	24		RW	Ort
ADK_CONTACTS_CODE_OF_ COUNTRY	16	C	4		RW	Landskod
ADK_CONTACTS_COUNTRY	17	C	24		RW	Land
ADK_CONTACTS_NOTES	18	C	1000		RW	Anteckningar
ADK_CONTACTS_ACCEPT_C OMMERCIAL_EMAIL	19	L	1		RW	Accepterar Epost (reklam)
ADK_CONTACTS_ACCEPT_C OMMERCIAL_SMS	20	L	1		RW	Accepterar SMS (reklam)
ADK_CONTACTS_CATEGORY	21	C	4		RW	Kundkategori
ADK_CONTACTS_DELETE_IN CLUDE_MAIN_REF	22	L	1		RW	Radera även om upplagd som referens på kund och/eller leverantör
ADK_CONTACTS_TIMESTAM P	23	Date	10		R	Tidsstämpel

Kontakttitlar – ADK_DB_CODE_OF_CONTACT_TITLES

Program: Visma Administration 500/1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_CONTACT_TI TLE_ID	0	D	9	0	RW	Titel ID
ADK_CODE_OF_CONTACT_TI TLE_NAME	1	C	50		RW	Namn
ADK_CODE_OF_CONTACT_TI TLE_TIMESTAMP	2	Date	10		R	Tidsstämpel

Kontaktgrupper – ADK_DB_CODE_OF_CONTACT_GROUPS

Program: Visma Administration 500/1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_CONTACT_G ROUPS_ID	0	D	9	0	RW	Grupp ID
ADK_CODE_OF_CONTACT_G ROUPS_NAME	1	C	50		RW	Namn
ADK_CODE_OF_CONTACT_G ROUPS_TIMESTAMP	2	Date	10		R	Tidsstämpel

Kontaktkopplingar – ADK_DB_CODE_OF_CONTACT_GROUP_CONTACTS *

Program: Visma Administration 500/1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_CONTACT_GROUP_CONTACTS_CONTACT_ID	0	D	9	0	RW	Kontakt ID
ADK_CODE_OF_CONTACT_GROUP_CONTACTS_GROUP_ID	1	D	9	0	RW	Grupp ID
ADK_CODE_OF_CONTACT_GROUP_CONTACTS_GROUP_NAME	2	C	50		R	Gruppnamn
ADK_CODE_OF_CONTACT_GROUP_CONTACTS_TIMESTAMP	3	Date	10		R	Tidsstämpel

* Kontaktkopplingsregistret används för att koppla kontakter till urvalsgrupper.

Kontoplan – ADK_DB_ACCOUNT

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_ACCOUNT_SEASON_ID	0	C	1		RW	Årsid, 'A' – 'J'
ADK_ACCOUNT_NUMBER	1	C	6		RW	Kontonummer, förberett för kontonummer med upp till sex siffror.
ADK_ACCOUNT_TEXT	2	C	59		RW	Kontobenämning
ADK_ACCOUNT_VAT_CODE	3	C	8		RW	Innehåller momsrapportkod.
ADK_ACCOUNT_SRU_CODE	4	C	5		RW	SRU-kod. Riksskatteverkets koder som används vid SRU – överföring via diskett. Används även vid export på SIE-format.
ADK_ACCOUNT_DEBIT_CREDIT	5	C	1		RW	Anger vanligaste konteringsätt: 'D' = Debet, 'K' = Kredit, '**' eller '=' = lika vanligt.
ADK_ACCOUNT_AUTOMATIC_DIVIDE	6	C	6		RW	Kod för eventuell automatfördelning.
ADK_ACCOUNT_MANUAL_CODING	7	L	1		RW	Sann innebär spärrad för manuell kontering. Används på konton som försystem använder t ex konto för kundfordringar som används av kundreskontra.
ADK_ACCOUNT_PROFIT_CENTRE_ON_ACCOUNT	8	C	1		RW	Resultatenhet på detta konto (A = alltid. E = Ej tillåtet. O = Ofta. S = Sällan).
ADK_ACCOUNT_PROFIT_CENTRE	9	C	6		RW	Resultatenhet. Definieras i eget register.
ADK_ACCOUNT_PROJECT_ON_ACCOUNT (4)	10	C	1		RW	Projekt på detta konto (A = alltid. E = Ej tillåtet. O = Ofta. S = Sällan).
ADK_ACCOUNT_PROJECT (4)	11	C	10		RW	Projekt. Definieras i eget register.
ADK_ACCOUNT_SUB_ACCOUNT (1)	12	C	1		RW	Subkonto på detta konto (A = alltid. E = Ej tillåtet. O = Ofta. S = Sällan).
ADK_ACCOUNT_SUB_ACCOUNT_DEFINITION (1)	13	C	2		RW	Subkonto
ADK_ACCOUNT_QUANTITY_UNIT_ON_ACCOUNT	14	C	1		RW	Kvantitetsenhet på detta konto (A = alltid. E = Ej tillåtet. O = Ofta. S = Sällan).
ADK_ACCOUNT_QUANTITY_UNIT	15	C	3		RW	Kvantitetsenhet.
ADK_ACCOUNT_QUANTITY_UNIT_NORMAL	16	C	3		RW	Kvantitetsenhet. Om skilt från blankt ska normalt kvantitet registreras på kontot.
ADK_ACCOUNT_ROW_INFO_ON_ACCOUNT	17	C	1		RW	Radinfo på detta konto (A = alltid. E = Ej tillåtet. O = Ofta. S = Sällan).
ADK_ACCOUNT_ROW_INFO	18	C	40		RW	Radinfo.
ADK_ACCOUNT_TYPE_OF_ACCOUNT	19	C	1		RW	Kontotyp på detta konto. Giltiga alternativ: T – tillgång, S – skuld, I – intäkt, K – kostnad.
ADK_ACCOUNT_NOT_ACTIVE	20	L	1		RW	Sann innebär att ett konto är inaktivt.

ADK_ACCOUNT_NEVER_DET AILED	21	L	1		RW	Sann innebär att kontot aldrig visas detaljerat vid utskrift av bokföringsunderlag.
ADK_ACCOUNT_TIMESTAMP	22	Date	10		R	Tidsstämpel
ADK_ACCOUNT_SALDO	23	D	14	2	R	Saldo
ADK_ACCOUNT_BALANCE	24	D	14	2	R	Ingående balans

(1) Fältet används endast i Visma Administration 2000.

(4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

Bokföringsår – ADK_DB_CODE_OF_BOOKINGYEAR

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_BOOKINGYE AR_ID	0	C	1		R	Års id
ADK_CODE_OF_BOOKINGYE AR_START	1	Date	8		R	Startdatum
ADK_CODE_OF_BOOKINGYE AR_END	2	Date	8		R	Slutdatum
ADK_CODE_OF_BOOKINGYE AR_TIMESTAMP	3	Date	10		R	Tidsstämpel
ADK_CODE_OF_BOOKINGYE AR_APTYPE	4	C	8		R	Kontoplanstyp
ADK_CODE_OF_BOOKINGYE AR_LOCKED_MONTH	5	C	6		R	Låst månad. Bokföringen är låst till och med månad AAAA-MM (tex 201502). Blank sträng innebär att ingen månad låsts i aktuellt år.
ADK_CODE_OF_BOOKINGYE AR_YEAR_LOCKED	6	L	1		R	Året är låst för bokföring.

Skattereduktion (Husarbete) – ADK_DB_TAX_REDUCTION

Program: Visma Administration 500/1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_TAX_REDUCTION_DOC UMENT_CONNECTION_NUMB ER	0	D	15	0	RW	Fakturanummer på fakturan vilken personen ska kopplas till
ADK_TAX_REDUCTION_PERS ONAL_IDENTITY_NUMBER	1	C	14		RW	Personnummer
ADK_TAX_REDUCTION_NAM E	2	C	50		RW	Namn (på sökande)
ADK_TAX_REDUCTION_APPL IED_AMOUNT	3	D	14	2	RW	Belopp skattereduktion
ADK_TAX_REDUCTION_APPL IED_DATE	4	Date	8		R	Ansökningsdatum
ADK_TAX_REDUCTION_GRA NTED_AMOUNT	5	D	14	2	R	Beviljat belopp
ADK_TAX_REDUCTION_GRA NTED_DATE	6	Date	8		R	Beviljat datum
ADK_TAX_REDUCTION_CUST OMER_PAYMENT_DATE	7	Date	8		R	Kundens betalningsdag
ADK_TAX_REDUCTION_STAT US	8	C	1		R	Status
ADK_TAX_REDUCTION_TIME STAMP	9	Date	10		R	Tidsstämpel
ADK_TAX_REDUCTION_DOC TYPE	10	C	1		RW	Dokumenttyp ('F' = Faktura, 'A' = Avtal, 'O' = Order).

Skattereduktion (Husarbete) på order – ADK_DB_TAX_REDUCTION_ORDER

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
----------	----	-----	-------	-----	------	-------------

ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMBER	0	D	15	0	RW	Ordernummer på ordern vilken personen ska kopplas till
ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER	1	C	14		RW	Personnummer
ADK_TAX_REDUCTION_NAME	2	C	50		RW	Namn (på sökande)
ADK_TAX_REDUCTION_APPLIED_AMOUNT	3	D	14	2	RW	Belopp skattereduktion
ADK_TAX_REDUCTION_APPLIED_DATE	4	Date	8		R	Ansökningsdatum
ADK_TAX_REDUCTION_GRANTED_AMOUNT	5	D	14	2	R	Beviljat belopp
ADK_TAX_REDUCTION_GRANTED_DATE	6	Date	8		R	Beviljat datum
ADK_TAX_REDUCTION_CUSTOMER_PAYMENT_DATE	7	Date	8		R	Kundens betalningsdag
ADK_TAX_REDUCTION_STATUS	8	C	1		R	Status
ADK_TAX_REDUCTION_TIMESTAMP	9	Date	10		R	Tidsstämpel
ADK_TAX_REDUCTION_DOCUMENT_TYPE	10	C	1		RW	Dokumenttyp ('F' = Faktura, 'A' = Avtal, 'O' = Order).

Skattereduktion (Husarbete) på avtal – ADK_DB_TAX_REDUCTION_AVTAL

Program: Visma Administration 1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_TAX_REDUCTION_DOCUMENT_CONNECTION_NUMBER	0	D	15	0	RW	Avtalsnummer på avtalet vilken personen ska kopplas till
ADK_TAX_REDUCTION_PERSONAL_IDENTITY_NUMBER	1	C	14		RW	Personnummer
ADK_TAX_REDUCTION_NAME	2	C	50		RW	Namn (på sökande)
ADK_TAX_REDUCTION_APPLIED_AMOUNT	3	D	14	2	RW	Belopp skattereduktion
ADK_TAX_REDUCTION_APPLIED_DATE	4	Date	8		R	Ansökningsdatum
ADK_TAX_REDUCTION_GRANTED_AMOUNT	5	D	14	2	R	Beviljat belopp
ADK_TAX_REDUCTION_GRANTED_DATE	6	Date	8		R	Beviljat datum
ADK_TAX_REDUCTION_CUSTOMER_PAYMENT_DATE	7	Date	8		R	Kundens betalningsdag
ADK_TAX_REDUCTION_STATUS	8	C	1		R	Status
ADK_TAX_REDUCTION_TIMESTAMP	9	Date	10		R	Tidsstämpel
ADK_TAX_REDUCTION_DOCUMENT_TYPE	10	C	1		RW	Dokumenttyp ('F' = Faktura, 'A' = Avtal, 'O' = Order).

Verifikationer – ADK_DB_VERIFICATION_HEAD

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_VERIFICATION_HEAD_YEAR_ID	0	C	1		RW	Års id (kan hämtas från ADK_DB_CODE_OF_BOOKINGYEAR)
ADK_VERIFICATION_HEAD_VER_NR	1	D	9	0	R	Vernr
ADK_VERIFICATION_HEAD_DATE1	2	Date	8		RW	Transaktionsdatum
ADK_VERIFICATION_HEAD_DATE2	3	Date	8		R	Registreringsdatum
ADK_VERIFICATION_HEAD_DATE3	4	Date	8		R	Senaste ändringsdatum

ADK_VERIFICATION_HEAD_I NFO	5	C	60		RW	Verifikationstext
ADK_VERIFICATION_HEAD_D EBIT	6	D	15	2	R	Debet
ADK_VERIFICATION_HEAD_C REDIT	7	D	15	2	R	Kredit
ADK_VERIFICATION_HEAD_D EBIT_R	8	D	15	2	R	Debet (kontoklass 1-8)
ADK_VERIFICATION_HEAD_C REDIT_R	9	D	15	2	R	Kredit (kontoklass 1-8)
ADK_VERIFICATION_HEAD_S IGN	10	C	8		R	Signatur
ADK_VERIFICATION_HEAD_K SIGN	11	C	10		R	Ändringssignatur
ADK_VERIFICATION_HEAD_C ORRECTS	12	D	9	0	R	Korrigerar
ADK_VERIFICATION_HEAD_C ORRECTED	13	D	9	0	R	Korrigerad av
ADK_VERIFICATION_HEAD_R OWS	14	Data	1		RW	Radpekare
ADK_VERIFICATION_HEAD_N ROWS	15	D	9	0	RW	Antal rader
ADK_VERIFICATION_HEAD_V ER_SERIE	16	C	1		RW	Verifikationsserie
ADK_VERIFICATION_HEAD_T YPE_OF_JOURNAL	17	C	1		R	Journaltyp
ADK_VERIFICATION_HEAD_J OURNAL_NR	18	D	15	0	R	Journalnummer
ADK_VERIFICATION_HEAD_T IMESTAMP	19	Date	10		R	Tidsstämpel

Verifikationsrader – ADK_DB_VERIFICATION_ROW

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_VERIFICATION_ROW_A AR_ID	0	C	1		R	Års id
ADK_VERIFICATION_ROW_V ER_NR	1	D	9	0	R	Verifikationsnummer
ADK_VERIFICATION_ROW_R OW	2	D	9	0	R	Radnummer
ADK_VERIFICATION_ROW_D ATE1	3	Date	8		R	Transaktionsdatum
ADK_VERIFICATION_ROW_D ATE2	4	Date	8		R	Ändringsdatum
ADK_VERIFICATION_ROW_A MOUNT	5	D	15	2	RW	Belopp
ADK_VERIFICATION_ROW_A DDED	6	L	1		R	Tillagd
ADK_VERIFICATION_ROW_D ELETED	7	L	1		R	Struken
ADK_VERIFICATION_ROW_A CCOUNT	8	C	6		RW	Konto
ADK_VERIFICATION_ROW_IN FO	9	C	60		RW	Info/text
ADK_VERIFICATION_ROW_P ROJECT (4)	10	C	10		RW	Projekt
ADK_VERIFICATION_ROW_P ROFIT_CENTRE	11	C	6		RW	Resultatenhet
ADK_VERIFICATION_ROW_K SIGN	12	C	10		R	Ändringssignatur
ADK_VERIFICATION_ROW_S UB_ACCOUNT (1)	13	C	2		RW	Subkonto
ADK_VERIFICATION_ROW_Q UANTITY	14	D	12	3	RW	Kvantitet
ADK_VERIFICATION_ROW_V ER_SERIE	15	C	1		R	Verifikationsserie

(1) Fältet används endast i Visma Administration 2000.

(4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

Landskoder – ADK_DB_CODE_OF_COUNTRY_CODE

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_COUNTRY_CODE_CODE	0	C	4		RW	Landskod
ADK_CODE_OF_COUNTRY_CODE_COUNTRY	1	C	50		RW	Land
ADK_CODE_OF_COUNTRY_CODE_ISO	2	L	1		R	ISO standard Sant / falskt
ADK_CODE_OF_COUNTRY_CODE_TIMESTAMP	3	Date	10		R	Tidsstämpel

Rabattkoder – ADK_DB_CODE_OF_DISCOUNT_CODE

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_DISCOUNT_FILTER	0	C	16		RW	Filter
ADK_CODE_OF_DISCOUNT_CODE	1	C	10		RW	Rabattkod
ADK_CODE_OF_DISCOUNT_PER_CENT	2	D	5	1	RW	Procent
ADK_CODE_OF_DISCOUNT_TIMESTAMP	3	Date	10		R	Tidsstämpel

Kundinbetalningar – ADK_DB_CUSTOMERPAYMENT

Program: Visma Administration 500/1000/2000, Visma Förening

Detta register visar samtliga betalningsrader, däremot går enbart direktbetalningar att skapa via API:t. Vissa av fälten används således inte vid direktbetalningar utan via exempelvis buntinbetalningar.

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CUSTOMERPAYMENT_NR	0	D	15	0	R	Buntnummer
ADK_CUSTOMERPAYMENT_INVOICE_NUMBER	1	D	15	0	RW	Fakturanummer
ADK_CUSTOMERPAYMENT_DATE	2	Date	8		RW	Betaldatum
ADK_CUSTOMERPAYMENT_ADJUSTMENT_FL	3	L	1		R	Bortskrivning
ADK_CUSTOMERPAYMENT_ADJUSTMENT_DUE_TO_CURRENCY	4	L	1		R	Kursdiff
ADK_CUSTOMERPAYMENT_FEE	5	L	1		R	Avgift
ADK_CUSTOMERPAYMENT_CURRENCY_RATE	6	D	12	6	RW	Valutakurs
ADK_CUSTOMERPAYMENT_AMOUNT_CURRENCY	7	D	14	2	RW	Belopp i valuta
ADK_CUSTOMERPAYMENT_AMOUNT_KR	8	D	14	2	R	Belopp som konteras betalningskontot
ADK_CUSTOMERPAYMENT_PAYTEXT	9	C	45		R	Betaltext
ADK_CUSTOMERPAYMENT_CURRENCY_CODE	10	C	4		R	Valutakod
ADK_CUSTOMERPAYMENT_PAYMENT_CODE	11	C	8		RW	Betalsätt
ADK_CUSTOMERPAYMENT_STATUS	12	C	60		R	Kommentar

ADK_CUSTOMERPAYMENT_VERID	13	C	10		R	Verifikationsnummer
ADK_CUSTOMERPAYMENT_INCLUDED_IN_PAYMENT_BATCH	14	L	1		R	Tillhör betalbunt
ADK_CUSTOMERPAYMENT_JOURNAL_NUMBER	15	D	15	0	R	Journalnr
ADK_CUSTOMERPAYMENT_VER_NUMBER	16	D	9	0	R	Vernr
ADK_CUSTOMERPAYMENT_VER_SERIES	17	C	1		R	Verserie
ADK_CUSTOMERPAYMENT_AMOUNT_OUT	18	D	14	2	R	Belopp
ADK_CUSTOMERPAYMENT_AMOUNT	19	D	14	2	RW	Belopp inhemsk valuta
ADK_CUSTOMERPAYMENT_ADJUSTMENT_CODE	20	C	8		RW	Bortskrivningskod
ADK_CUSTOMERPAYMENT_ADJUSTMENT	21	D	14	2	R	Bortskrivningsbelopp
ADK_CUSTOMERPAYMENT_ADJUSTMENT_KR	22	D	14	2	R	Bortskrivningsbelopp kr
ADK_CUSTOMERPAYMENT_ACCOUNT	23	C	6		RW	Kontonummer, RW endast på avgiftsrader
ADK_CUSTOMERPAYMENT_CANCELLED	24	L	1		R	Makulerad
ADK_CUSTOMERPAYMENT_ROW_NUMBER	25	D	9	0	RW	Radnr
ADK_CUSTOMERPAYMENT_CONNECTION_ROW	26	D	9	0	RW	Kopplad rad
ADK_CUSTOMERPAYMENT_FILE_FL	27	L	1		R	Inläst från återrapportering
ADK_CUSTOMERPAYMENT_NOT_DONE	28	L	1		R	Ej klar för journalföring
ADK_CUSTOMERPAYMENT_INVOICE_REF	29	C	30		R	Fakturareferens
ADK_CUSTOMERPAYMENT_STATUS_C	30	C	1		R	Status
ADK_CUSTOMERPAYMENT_SETTLED	31	L	1		R	Kvittning
ADK_CUSTOMERPAYMENT_SETTLED_INVOICE_NR	32	D	15	0	R	Fakturanummer vid kvittning
ADK_CUSTOMERPAYMENT_AMOUNT_FEE	33	D	14	2	R	Avgiftsbelopp
ADK_CUSTOMERPAYMENT_AMOUNTDEMAND	34	D	14	2	R	Belopp (Räknar bort kursdiffar och bortskrivningar)
ADK_CUSTOMERPAYMENT_AMOUNT_CANCELLED	35	D	14	2	R	Makulerat belopp
ADK_CUSTOMERPAYMENT_AMOUNT_PAID	36	D	14	2	R	Betalt belopp
ADK_CUSTOMERPAYMENT_ROW_TYPE	37	C	1		RW	Radtyp – H, A, B, K
ADK_CUSTOMERPAYMENT_TIMESTAMP	38	Date	10		R	Tidsstämpel

Radtypen (fältid 37) bestämmer hur de skrivbara fälten används. Anges ingen radtyp antas det vara av typ H. Följande radtyper finns

- H – *Huvudrad*, direktbetalningens huvudrad, betalsätt måste anges
- A – *Avgiftsrad*, huvudradens radnummer måste kopplas, enbart belopp i kronor
- B – *Bortskrivningsrad*, huvudradens radnummer måste kopplas, bortskrivningskod måste anges
- K – *Kursdiffrad* (går ej att skapa, skapas automatiskt om kursdiff finns)

För att skapa en inbetalning ska alltså en huvudrad först skapas. Radtypen sätts med hjälp av fältet ADK_CUSTOMERPAYMENT_ROW_TYPE, anges inget antas det vara en huvudrad. Samtliga rader som skapas får automatiskt ett radnummer, men för radtyperna A (avgiftsrader) och B (bortskrivningar) måste man sätta fältet ADK_CUSTOMERPAYMENT_CONNECTION_ROW till huvudradens radnummer (ADK_CUSTOMERPAYMENT_ROW_NUMBER). Rader av typen K (kursdiffar) skapas per automatik och kan inte påverkas av användaren.

Bortskrivningskoder – ADK_DB_CODE_OF_ADJUSTMENT_CODE

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CODE_OF_ADJUSTMENT_CODE_CODE	0	C	8		RW	Kod
ADK_CODE_OF_ADJUSTMENT_CODE_TEXT	1	C	20		RW	Text
ADK_CODE_OF_ADJUSTMENT_CODE_ACCOUNT	2	C	6		RW	Konto
ADK_CODE_OF_ADJUSTMENT_CODE_TIMESTAMP	3	Date	10		R	Tidsstämpel

Företagsinställningar – ADK_DB_PRM2

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_PRM2_VAT1	0	D	8	5	R	Momskod 1
ADK_PRM2_VAT2	1	D	8	5	R	Momskod 2
ADK_PRM2_VAT3	2	D	8	5	R	Momskod 3
ADK_PRM2_VAT4	3	D	8	5	R	Momskod 4
ADK_PRM2_NORM	4	C	1		R	Vanligaste momskod (normalkod)
ADK_PRM2_DEFAULT_VATCODE_CARGO_FEE	5	C	1		R	Momskod frakt (normalvärde)
ADK_PRM2_DEFAULT_VATCODE_DISPATCH_FEE	6	C	1		R	Momskod exp avg (normalvärde)
ADK_PRM2_DEFAULT_VATCODE_DEMAND_FEE	7	C	1		R	Momskod kravavg (normalvärde)
ADK_PRM2_LIMIT_DISPATCH_FEE	8	D	13	2	R	Gräns för exp avg
ADK_PRM2_LIMIT_CARGO_FEE	9	D	13	2	R	Gräns för frakt
ADK_PRM2_DEFAULT_PRICE_LIST	10	C	8		R	Prislista, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_EXPORT	11	L	1		R	Exportkund, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_EU_CUSTOMER	12	L	1		R	EU-kund, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_PAYMENT_TERMS	13	C	8		R	Betalningsvillkor, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_DELIVERY_TERMS	14	C	8		R	Leveransvillkor, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_DELIVERY_WAY	15	C	8		R	Leveranssätt, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_CREDIT_LIMIT	16	D	15	0	R	Kreditgräns, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_CATEGORY	17	C	4		R	Kundkategori, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_SELLER (2)	18	C	10		R	Säljare, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_DISTRICT (2)	19	C	8		R	Distrikt, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_INTEREST_INVOICE	20	L	1		R	Räntefaktura, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_REMINDER	21	L	1		R	Krav, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_DEMAND_FEE	22	L	1		R	Kravavgift, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_DO_NOT_CREATE_BACKORDER (1)	23	L	1		R	Skapa ej restorder, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_DISPATCH_FEE	24	L	1		R	Exp avg, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_SUMMARY_INVOICE (1)	25	L	1		R	Samlingsfaktura, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_CARGO_FEE	26	L	1		R	Frakt, kund (normalvärde)

ADK_PRM2_DEFAULT_CUST_DISPATCHER (3)	27	C	7		R	Speditör, kund (normalvärde)
ADK_PRM2_DEFAULT_CUST_ADDRESS_TO_BOOKING (1)	28	L	1		R	Adress på order till beställning, kund (normalvärde)
ADK_PRM2_DEFAULT_OOI_OR_REFERENCE_NAME	29	C	25		R	Vår referens, offert/order/faktura (normalvärde)
ADK_PRM2_DEFAULT_OOI_DISPATCH_FEE	30	D	10	2	R	Exp avg, offert/order/faktura (normalvärde)
ADK_PRM2_DEFAULT_OOI_CARGO_FEE	31	D	10	2	R	Frakt, offert/order/faktura (normalvärde)
ADK_PRM2_DEFAULT_OOI_CARGO_FEE_INCL_VAT	32	D	10	2	R	Frakt inkl moms, offert/order/faktura (normalvärde)
ADK_PRM2_DEFAULT_OOI_DISPATCH_FEE_INCL_VAT	33	D	10	2	R	Exp avg inkl moms, offert/order/faktura (normalvärde)
ADK_PRM2_DEFAULT_INVOICE_CREDIT_NOTE_STORAGE_AFFECT	34	L	1		R	Lagerpåverkan kreditfaktura (normalvärde)
ADK_PRM2_DEFAULT_OFFER_VALID_TIME (1)	35	D	4	0	R	Offertgiltighet (normalvärde)
ADK_PRM2_NEXT_CUSTOMER_NO	36	C	16		R	Nästa kundnr
ADK_PRM2_NEXT_SUPPLIER_NO	37	C	16		R	Nästa leverantörsnr
ADK_PRM2_NEXT_ARTICLE_NO	38	C	40		R	Nästa artikelnr
ADK_PRM2_DEFAULT_ACCOUNT_CASH	39	C	6		R	Kassakonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_PGRO	40	C	6		R	Plusgirokonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_BGIRO	41	C	6		R	Bankgirokonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_CUSTOMER_CLAIM	42	C	6		R	Kundfordrankonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_VAT_IN	43	C	6		R	Ingående momskonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_SUPPL_DEBT	44	C	6		R	Leverantörsskuldkonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_VAT1	45	C	6		R	Utg moms1 konto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_VAT2	46	C	6		R	Utg moms2 konto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_VAT3	47	C	6		R	Utg moms3 konto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_VAT4	48	C	6		R	Utg moms4konto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_SALES	49	C	6		R	Försäljningskonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_SALES_EXPORT	50	C	6		R	Exportförsäljningskonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_ROUND	51	C	6		R	Avrundningskonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_DISP_FEE	52	C	6		R	Exp avg konto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_DISP_FEE_EXPORT	53	C	6		R	Exp avg export konto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_CARGO_FEE	54	C	6		R	Frakt konto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_CARGO_FEE_EXP	55	C	6		R	Frakt export konto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_DISCOUNT	56	C	6		R	Rabattkonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_CUST_INTEREST	57	C	6		R	Räntekonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_DEMAND_FEE	58	C	6		R	Kravavgiftskonto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_DEMAND_FEE_EXPORT	59	C	6		R	Kravavg export konto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_SUP_INV_CARGO_FEE	60	C	6		R	Frakt på leverantörsfaktura konto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_DISCOUNT_EXPORT	61	C	6		R	Rabatt export konto (normalvärde)
ADK_PRM2_DEFAULT_ACCOUNT_CURRENCY_ADJ	62	C	6		R	Kursdifferens

ADK_PRM2_DEFAULT_ACCO UNT_SALES_EU	63	C	6		R	Försäljning EU konto (normalvärde)
ADK_PRM2_DEFAULT_ACCO UNT_DISP_FEE_EU	64	C	6		R	Exp avg EU konto (normalvärde)
ADK_PRM2_DEFAULT_ACCO UNT_CARGO_FEE_EU	65	C	6		R	Frakt EU konto (normalvärde)
ADK_PRM2_DEFAULT_ACCO UNT_DISCOUNT_EU	66	C	6		R	Given rabatt EU konto (normalvärde)
ADK_PRM2_DEFAULT_ACCO UNT_PAYMENT_FEE	67	C	6		R	Bankkostnader konto (normalvärde)
ADK_PRM2_DEFAULT_ACCO UNT_VAT_EU	68	C	6		R	Försäljning EU m moms konto (normalvärde)
ADK_PRM2_DEFAULT_ACCO UNT_DEBT_DELIV_NOTE (1)	69	C	6		R	Skuld ink följesedel konto (normalvärde)
ADK_PRM2_DEFAULT_MEMB ER_CATEGORY (5)	70	C	6		R	Medlemskategori (normalvärde)
ADK_PRM2_MEMBER_PNR_C HECK_ACTIVE (5)	71	L	1		R	Personnummerkontroll (aktiverad)
ADK_PRM2_FREE_CAT1_ACT IVE (5)	72	L	1		R	Fri kategori 1 (aktiverad)
ADK_PRM2_FREE_CAT2_ACT IVE (5)	73	L	1		R	Fri kategori 2 (aktiverad)
ADK_PRM2_FREE_CAT3_ACT IVE (5)	74	L	1		R	Fri kategori 3 (aktiverad)
ADK_PRM2_FREE_CAT4_ACT IVE (5)	75	L	1		R	Fri kategori 4 (aktiverad)
ADK_PRM2_FREE_CAT5_ACT IVE (5)	76	L	1		R	Fri kategori 5 (aktiverad)
ADK_PRM2_FREE_CAT6_ACT IVE (5)	77	L	1		R	Fri kategori 6 (aktiverad)
ADK_PRM2_FREE_CAT7_ACT IVE (5)	78	L	1		R	Fri kategori 7 (aktiverad)
ADK_PRM2_FREE_CAT8_ACT IVE (5)	79	L	1		R	Fri kategori 8 (aktiverad)
ADK_PRM2_FREE_CAT9_ACT IVE (5)	80	L	1		R	Fri kategori 9 (aktiverad)
ADK_PRM2_FREE_CAT10_AC TIVE (5)	81	L	1		R	Fri kategori 10 (aktiverad)
ADK_PRM2_FREE_CAT1_NA ME (5)	82	C	10		R	Namn på fri kategori 1
ADK_PRM2_FREE_CAT2_NA ME (5)	83	C	10		R	Namn på fri kategori 2
ADK_PRM2_FREE_CAT3_NA ME (5)	84	C	10		R	Namn på fri kategori 3
ADK_PRM2_FREE_CAT4_NA ME (5)	85	C	10		R	Namn på fri kategori 4
ADK_PRM2_FREE_CAT5_NA ME (5)	86	C	10		R	Namn på fri kategori 5
ADK_PRM2_FREE_CAT6_NA ME (5)	87	C	10		R	Namn på fri kategori 6
ADK_PRM2_FREE_CAT7_NA ME (5)	88	C	10		R	Namn på fri kategori 7
ADK_PRM2_FREE_CAT8_NA ME (5)	89	C	10		R	Namn på fri kategori 8
ADK_PRM2_FREE_CAT9_NA ME (5)	90	C	10		R	Namn på fri kategori 9
ADK_PRM2_FREE_CAT10_NA ME (5)	91	C	10		R	Namn på fri kategori 10

(1) Fältet används endast i Visma Administration 2000.

(2) Fältet används endast i Visma Administration 1000/2000.

(3) Fältet används endast i Visma Administration 500/1000/2000.

(5) Fältet används endast i Visma Förening

Utbetalningar – ADK_DB_SUPPLIERPAYMENT

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_SUPPLIERPAYMENT_PA YORDER	0	D	15	0	R	Betalnr
ADK_SUPPLIERPAYMENT_PA YROW	1	D	9	0	R	Radnr

ADK_SUPPLIERPAYMENT_N R	2	D	15	0	R	Leverantörbetalningnummer
ADK_SUPPLIERPAYMENT_TY PE	3	C	2		R	Typ
ADK_SUPPLIERPAYMENT_SU BTYPE	4	C	1		R	Undertyp
ADK_SUPPLIERPAYMENT_IN VOICE_NR	5	D	15	0	R	Leverantörsfakturanummer
ADK_SUPPLIERPAYMENT_DA TE	6	Date	8		R	Betaldatum
ADK_SUPPLIERPAYMENT_EX PDATE	7	Date	8		R	Förfalldatum
ADK_SUPPLIERPAYMENT_A MOUNT_V	8	D	14	2	R	Belopp valuta
ADK_SUPPLIERPAYMENT_A MOUNT_I	9	D	14	2	R	Belopp inhemsk
ADK_SUPPLIERPAYMENT_IN FO	10	C	25		R	Info
ADK_SUPPLIERPAYMENT_O CR	11	C	25		R	OCR/Referensnummer
ADK_SUPPLIERPAYMENT_SA CCOUNT	12	C	6		R	Skuldkonto
ADK_SUPPLIERPAYMENT_UA CCOUNT	13	C	6		R	Utbetalningskonto
ADK_SUPPLIERPAYMENT_R OW	14	D	9	0	R	Rad
ADK_SUPPLIERPAYMENT_JO URNAL	15	D	15	0	R	Journalnummer
ADK_SUPPLIERPAYMENT_VE RNR	16	D	9	0	R	Vernummer
ADK_SUPPLIERPAYMENT_PA YED	17	L	1		R	Betald
ADK_SUPPLIERPAYMENT_CA NCELLED	18	L	1		R	Makulerad
ADK_SUPPLIERPAYMENT_C ON_SPEC	19	D	15	0	R	Kopplar avgiftsrader och kursdiffnader till betalrader, unika betalningsnummret
ADK_SUPPLIERPAYMENT_PA YMENT_CODE	20	D	4	0	R	Betalsätt
ADK_SUPPLIERPAYMENT_SU PPLIER_NR	21	C	16		R	Leverantörsnummer
ADK_SUPPLIERPAYMENT_SU PPLIER_NAME	22	C	50		R	Leverantörsnamn
ADK_SUPPLIERPAYMENT_SU PPLIER_CUSTOMER_NR	23	C	20		R	Kundnr
ADK_SUPPLIERPAYMENT_SU PPLIER_ADR1	24	C	35		R	Postadress
ADK_SUPPLIERPAYMENT_SU PPLIER_ADR2	25	C	35		R	Postadr 2
ADK_SUPPLIERPAYMENT_SU PPLIER_ZIPCODE	26	C	12		R	Postnr
ADK_SUPPLIERPAYMENT_SU PPLIER_CITY	27	C	24		R	Ort
ADK_SUPPLIERPAYMENT_SU PPLIER_COUNTRY	28	C	24		R	Land
ADK_SUPPLIERPAYMENT_C URRENCY_CODE	29	C	4		R	Valutakod
ADK_SUPPLIERPAYMENT_AC COUNT_NR	30	C	34		R	Kontonummer
ADK_SUPPLIERPAYMENT_C URRENCY_RATE	31	D	12	6	R	Valutakurs
ADK_SUPPLIERPAYMENT_KA MOUNT_V	32	D	14	2	R	Belopp
ADK_SUPPLIERPAYMENT_AA MOUNT_V	33	D	14	2	R	Belopp
ADK_SUPPLIERPAYMENT_AA MOUNT_I	34	D	14	2	R	Belopp
ADK_SUPPLIERPAYMENT_C URRENCY_ACCOUNT	35	C	6		R	Kursdiffkonto
ADK_SUPPLIERPAYMENT_FE E_CODE	36	C	1		R	Avgiftskod
ADK_SUPPLIERPAYMENT_FE E_AMOUNT	37	D	14	2	R	Avgiftsbelopp
ADK_SUPPLIERPAYMENT_FE E_ACCOUNT	38	C	6		R	Avgiftskonto

ADK_SUPPLIERPAYMENT_VE R_SERIE	39	C	1		R	Verserie
ADK_SUPPLIERPAYMENT_PA YDATE	40	Date	8		R	Betaldatum
ADK_SUPPLIERPAYMENT_A MOUNT	41	D	14	2	R	Betalförslag
ADK_SUPPLIERPAYMENT_IN VNR	42	C	25		R	Fakturanummer
ADK_SUPPLIERPAYMENT_P GIRO	43	C	13		R	Plusgiro
ADK_SUPPLIERPAYMENT_B GIRO	44	C	34		R	Bankgiro
ADK_SUPPLIERPAYMENT_VE R_ID	45	C	10		R	Vernr
ADK_SUPPLIERPAYMENT_ST ATUS	46	C	25		R	Status
ADK_SUPPLIERPAYMENT_PA YTXT	47	C	25		R	Betaltext
ADK_SUPPLIERPAYMENT_TI MESTAMP	48	C	10		R	Tidsstämpel

Verifikationsserier – ADK_DB_VERIFICATION_SERIES

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_VERIFICATION_SERIES _VER_SERIE	0	C	1		R	Verifikationsserie
ADK_VERIFICATION_SERIES _NAME	1	C	25		R	Namn
ADK_VERIFICATION_SERIES _ACTIVE	2	L	1		R	Verifikationsserien aktiv
ADK_VERIFICATION_SERIES _USERS	3	C	60		R	Användare
ADK_VERIFICATION_SERIES _TIMESTAMP	4	Date	10		R	Tidsstämpel

Bokföringshistorik – ADK_DB_BOOKKEEPINGHIST

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_BOOKKEEPINGHIST_NR	0	D	15	0	R	Löpnr
ADK_BOOKKEEPINGHIST_DA TE	1	Date	8		R	Registreringsdatum
ADK_BOOKKEEPINGHIST_TI ME	2	D	9	0	R	Tid (i sekunder från dygnets början)
ADK_BOOKKEEPINGHIST_SI GN	3	C	8		R	Signatur
ADK_BOOKKEEPINGHIST_TE XT	4	C	100		R	Text
ADK_BOOKKEEPINGHIST_PR INTED	5	L	1		R	Utskriven
ADK_BOOKKEEPINGHIST_TI MESTAMP	6	Date	10		R	Tidsstämpel

Periodiseringar – ADK_DB_PERIODIC_ADJUSTMENT (a)

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_PERIODIC_ADJUSTMEN T_ID	0	C	6		R	Id
ADK_PERIODIC_ADJUSTMEN T_TEXT	1	C	60		RW	Text
ADK_PERIODIC_ADJUSTMEN T_ACCOUNT	2	C	6		RW	Konto i)

ADK_PERIODIC_ADJUSTMENT_DATE	3	Date	8		RW	Datum b)
ADK_PERIODIC_ADJUSTMENT_REMAINING_AMOUNT	4	D	14	2	R	Kvar att periodisera
ADK_PERIODIC_ADJUSTMENT_NUMBER_DONE	5	D	4	0	R	Antal genomförda periodiseringar
ADK_PERIODIC_ADJUSTMENT_START_DATE	6	Date	8		RW	Startdatum d)
ADK_PERIODIC_ADJUSTMENT_END_DATE	7	Date	8		R	Slutdatum
ADK_PERIODIC_ADJUSTMENT_TOTAL_AMOUNT	8	D	14	2	R	Totalt belopp
ADK_PERIODIC_ADJUSTMENT_DONE_AMOUNT	9	D	14	2	R	Genomfört belopp
ADK_PERIODIC_ADJUSTMENT_AMOUNT_TO_DO	10	D	14	2	R	Periodens belopp att periodisera
ADK_PERIODIC_ADJUSTMENT_CONTRA_ACCOUNT	11	C	6		RW	Motkonto h)
ADK_PERIODIC_ADJUSTMENT_NUMBER_OF_MONTHS	12	D	4	0	RW	Antal månader d)
ADK_PERIODIC_ADJUSTMENT_PROJECT (4)	13	C	10		RW	Projekt
ADK_PERIODIC_ADJUSTMENT_PROFIT_CENTRE	14	C	6		RW	Resultatenhet
ADK_PERIODIC_ADJUSTMENT_VER_SERIE	15	C	1		RW	Verifikationsserie för periodiseringen f)
ADK_PERIODIC_ADJUSTMENT_TYPE	16	D	4	0	RW	Periodiseringstyp d) g)
ADK_PERIODIC_ADJUSTMENT_CONNECTED_DOCUMENT	17	D	15	0	RW	Kopplat dokumentnr/vernr d)
ADK_PERIODIC_ADJUSTMENT_CONNECTED_VER_SERIE	18	C	1		RW	Kopplad verserie (endast verifikation) e) f)
ADK_PERIODIC_ADJUSTMENT_SHOW_DONE	19	L	1		RW	Visa genomförda periodiseringar c)
ADK_PERIODIC_ADJUSTMENT_NAME	20	C	20		RW	Namn
ADK_PERIODIC_ADJUSTMENT_CONNECTED_ROW_NUMBER	21	D	9	0	RW	Kopplad rad d)
ADK_PERIODIC_ADJUSTMENT_CONNECTED_BOOKINGYEAR_ID	22	C	1		RW	Kopplat årsid (endast verifikation) e)

(4) Fältet används endast i Visma Administration 1000/2000 och Visma Förening.

a) Tabellen presenterar poster på liknande sätt som utskriften Periodiseringar – Sammanställning i programmet. Man kan filtrera på datum och om helt genomförda periodiseringar ska visas. Man kan även skapa periodiseringar, men inte ändra befintliga.

b) ADK_PERIODIC_ADJUSTMENT_DATE är det datum man vill ha periodiseringarna beräknade för och styr vilka värden man får i genomfört belopp, kvar att periodisera samt periodens belopp att periodisera. (En period är alltid en månad). Används inte vid skapande av periodisering.

c) Om ADK_SHOW_PERIODIC_ADJUSTMENT_DONE sätts till TRUE visas även helt genomförda periodiseringar. Lämnas fältet tomt visas endast de periodiseringar som har belopp kvar att periodisera på det datum man angett i fältet ADK_PERIODIC_ADJUSTMENT_DATE. Används inte vid skapande av periodisering.

d) Obligatoriskt fält vid skapande av periodisering.

e) Obligatoriskt fält vid skapande av periodisering av verifikationsrad.

f) Obligatoriskt endast om verifikationsserie är aktiverat i företaget.

g) Tillåtna värden finns i enum ADK_PERIOD_TYPE. Se nedan.

h) Obligatoriskt fält vid skapande av periodisering av typen PERIOD_LEVFKT och PERIOD_VERIF. Ej tillåtet att ange för andra typer.

i) Obligatoriskt fält vid skapande av periodisering av typen PERIOD_LEVFKTNY. Ej tillåtet att ange för andra typer.

ADK_PERIODIC_TYPE

```
typedef enum
```

```

{
    PERIOD_LEVFKT = 3,
    PERIOD_VERIF = 4,
    PERIOD_KUNDFKT = 5,
    PERIOD_LEVFKTNY = 6
} ADK_PERIOD_TYPE;

```

Kundartiklar – ADK_DB_CUSTOMER_ARTICLE

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_CUSTOMER_ARTICLE_ARTNR	0	C	40		RW	Artikelnr
ADK_CUSTOMER_ARTICLE_CUSTNR	1	C	16		RW	Kundnr
ADK_CUSTOMER_ARTICLE_CUSTARTNR	2	C	40		RW	Kundens artikelnr
ADK_CUSTOMER_ARTICLE_TIMESTAMP	3	Date	10		R	Tidsstämpel

Kopplade underlag – ADK_DB_ATTACHMENT_INFO

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_ATTACHMENT_INFO_GUID	0	C	60		R	Unikt id
ADK_ATTACHMENT_INFO_TYPE	1	C	2		R	Typ i Mobile scanner-tjänsten
ADK_ATTACHMENT_INFO_DOCUMENTNR	2	D	15	0	R	Levfakturanr (a)
ADK_ATTACHMENT_INFO_DOCUMENTTYPE	3	C	3		R	Dokumenttyp underlaget kopplats till (LF eller V)
ADK_ATTACHMENT_INFO_ACCOUNTID	4	C	1		R	Bokföringsårsid (b)
ADK_ATTACHMENT_INFO_VERIFICATIONNR	5	D	9	0	R	Verifikationsnr (b)
ADK_ATTACHMENT_INFO_VERIFICATIONSERIE	6	C	1		R	Verifikationsserie (b)
ADK_ATTACHMENT_INFO_TIMESTAMP	7	Date	10		R	Tidsstämpel

(a) Används endast för underlag kopplade till leverantörsfakturer

(b) Används endast för underlag kopplade till verifikationer

Husarbetstyper – ADK_DB_TAX_REDUCTION_TYPES

Program: Visma Administration 500/1000/2000

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_TAX_REDUCTION_TYPES_ID	0	C	4		R	Unikt id
ADK_TAX_REDUCTION_TYPES_NAME	1	C	60		R	Namn
ADK_TAX_REDUCTION_TYPES_TYPE	2	C	1		R	Typ (R = ROT, H = RUT, O = Övrigt, G = Grön teknik) (1)
ADK_TAX_REDUCTION_TYPES_SUBTYPE	3	C	1		R	Subtyp (A = Arbete, M = Material, O = Övrigt, S = Schablon, G = Grön teknik) (1)
ADK_TAX_REDUCTION_TYPES_ACTIVE	4	L	1		R	Aktiv (för framtida bakåtkompatibilitet)
ADK_TAX_REDUCTION_TYPES_TAXNAME	5	C	60		R	XML-nodens namn i SKVs format
ADK_TAX_REDUCTION_TYPES_SCHAB_PROC	6	D	6	2	R	Procentsatser för schablontyper och grön teknik

(1) Möjliga kombinationer för husarbetstyper

Typ R kan ha Subtyp A, M

Typ H kan ha subtyp A, M, S

Typ O kan ha subtyp O

Typ G kan ha subtyp G, O

Företagsinställningar – ADK_DB_PRM3

Program: Visma Administration 500/1000/2000, Visma Förening

Fältnamn	Id	Typ	Bredd	Dec	RW/R	Beskrivning
ADK_PRM3_DEFAULT_ACCO UNT_BUILDSERVICE (3)	0	C	6		R	Deafultkonto för omvänd skattskyldighet

(3) Fältet används endast i Visma Administration 500/1000/2000.

Beskrivning av felkoder

Här följer en beskrivning av alla de felkoder som kan genereras av **API:et**.
Därefter följer en beskrivning av de varningar som kan förekomma.

Felkoder (IProgramPart: APP (=4))

Kod	Felmeddelande	Förklaring
0	"Successful"	Inget fel har uppstått
1	"Internt fel, kontakta Visma SPCS"	Ett internt fel har uppstått.
2	"Ändring inte tillåten. Rabattavtal finns."	Ändring ej tillåten i något av fälten prislista, fakturarabatt eller radrabatt vid skapande/uppdaterande av order, offert, faktura eller kund om ett rabattavtal finns.
3	"Kan inte ändra EU- eller export-flagga på faktura som är bokförd som inhemsk faktura."	Kan uppstå vid arbete med en kundfaktura.
4	"Kan ej utföra önskad ändring."	Makulering inte tillåten om nästa dokument i kedjan order – offer – faktura är skapat. Kan även uppstå om en order är markerad ej klar. Då är det ej tillåtet att ändra utskriftsflaggorna ordererkännande, plocklista eller följesedel.
5	"Kan inte sätta en order, offert eller faktura som skickad om den är markerad som ej klar."	Kan uppstå vid arbete med order, offert eller faktura.
6	"Får inte ändras på exportfaktura."	Inklusivemoms flaggan får ej ändras om exportflaggan är satt till sann.
7	"Inte tillåtet med artikelnummer på textrad."	Artikelnummer får ej anges om radtypen i en order-, offert- eller fakturarad är satt till textrad.
8	"Det går inte att ändra artikelnummer på en paketartikel. (Vill du ta bort paketet måste du radera det.)"	Ej tillåtet att ändra artikelnumret på en paketartikel vid arbete med order-, offert- eller fakturarad.
9	"Ändring inte tillåten. Makulerad."	Ändring ej tillåten om dokumentet är makulerat.
10	"Ändring inte tillåten. Order skapad."	Ändringar på offert ej tillåten om order är skapad.
11	"Ändring inte tillåten. Faktura skapad."	Ändring på order är ej tillåten om faktura är skapad. Kan även uppstå vid avmakuleringen av en faktura om grundordern har skapat en ny faktura.
12	"Ändring inte tillåten. Skickad."	Ändring ej tillåten då dokumentet är utskrivet.
13	"Ändring inte tillåten. Ordern är markerad som levererad."	Ändring på order ej tillåten om ordern är markerad som levererad.
14	"En annan order är skapad för offerten. Kan inte upphäva makuleringen."	Vid avmakulering av order. Om grundofferten har skapat en ny order kan inte avmakulering utföras.
15	"Kunden är inte inställd för samlingsfakturerings."	Flaggan för samlingsfakturerings på en ordern kan ej sättas om kunden inte är inställd för samlingsfakturerings.
16	"Ändring inte tillåten. Ordern är redan restnoterad."	Ej tillåtet att ändra flaggan för restnotera ej om en order redan är restnoterad.
17	"Restorderflaggan kan inte ändras på en rad i paket. Paketartikeln styr om paketet kan restnoteras eller inte."	Kan uppstå vid arbete med en order.
18	"Ändring inte tillåten på deltotalrad."	Ändring ej tillåten när en order-, offert- eller fakturarad är satt som deltotalrad.
19	"Ändring inte tillåten. Raden är journalförd via lagerförändringsjournal."	Ändring ej tillåten då en order-, offert- eller fakturarad är journalförd.
20	"Kan ej hitta artikeln i artikelregistret"	Används ej
21	"Kan ej hitta kunden i kundregistret"	Används ej
22	"Resultatenhet är ej tillåtet att ange till aktuellt konto."	Kan uppstå vid arbete med en leverantörsfaktura.
23	"Ändring inte tillåten. Fakturan journalförd."	Ändring ej tillåtet om fakturan är journalförd. Kan vara kundfaktura eller leverantörsfaktura.
24	"Ändring inte tillåten. Fakturan är bokförd."	Ändring ej tillåtet om fakturan är bokförd. Kan vara kundfaktura eller leverantörsfaktura.

25	"Ändring inte tillåten. Fakturan ligger i reskontra."	Ändring ej tillåtet om fakturan ligger i reskontra. Kan vara kundfaktura eller leverantörsfaktura.
26	"Avmakulering kan inte göras. Det finns order som har skapat andra fakturor."	Kan uppstå vid avmakulering av samlingsfaktura.
27	"För att kunna makulera en faktura med ränterader måste du först manuellt radera ränteraderna. Därefter kan du makulera igen."	Kan uppstå vid radering av kundfakturor.
28	"Ändring av kundnummer på samlingsfaktura är inte tillåtet."	Ej tillåtet att ändra kundnummer på en kundfaktura som är markerad som samlingsfaktura.
29	"Rad med orderinformation får inte ändras."	Ej tillåtet att ändra på en fakturad som innehåller orderinformation.
30	"Rad med avtalsperiod får inte ändras."	Ej tillåtet att ändra på en fakturad som innehåller en avtalsperiod.
31	"Ej tillåtet att redigera ingående fakturor."	Kan uppstå vid arbete med en kundfaktura.
32	"Kalkylpriser går inte att ändra manuellt på en paketartikel. Paketartikelens kalkylpris är summan av ingående artiklars kalkylpriser."	Ej tillåtet att ändra något av kalkylfälten inköpspris, frakt eller övriga kostnader på en paketartikel.
33	"En paketartikel kan inte vara lagervara"	Ej tillåtet att försöka sätta flaggan för lagervara till sann på en paketartikel.
34	"Felaktig sträng."	En sträng som skickas till funktionen <code>AdkSetStr</code> får ej vara längre än fältlängden på det fält som skall anta strängen.
35	"Kan ej hitta angiven prislista."	Kan uppstå vid skapandet av artikel, kund, order, offert eller faktura då angiven prislista ej kan finnas i tabellen för prislistor.
36	"Kan inte skapa order. Offert kan inte läsas."	Används ej
37	"Kan ej hitta angiven kundkategori."	Kan uppstå vid skapande/uppdaterande av kund.
38	"Kan ej hitta angivet distrikt."	Kan uppstå vid skapande/uppdaterande av kund, order, offert eller faktura.
39	"Kan ej hitta angiven säljare."	Kan uppstå vid skapande/uppdaterande av kund, order, offert eller faktura.
40	"Kan ej hitta angivet leveransvillkor."	Kan uppstå vid skapande/uppdaterande av kund, leverantör, order, offert eller faktura.
41	"Kan ej hitta angivet betalvillkor."	Kan uppstå vid skapande/uppdaterande av kund, leverantör, order, offert eller faktura.
42	"Kan ej hitta angivet leveranssätt."	Kan uppstå vid skapande/uppdaterande av kund, leverantör, order, offert eller faktura.
43	"Kan ej hitta angiven valutakod."	Kan uppstå vid skapande/uppdaterande av kund, leverantör, order, offert, faktura. Eller leverantörsfaktura.
44	"Kan ej hitta angiven språkkod."	Kan uppstå vid skapande/uppdaterande av kund, leverantör, order, offert eller faktura.
45	"Kan ej hitta angiven artikelgrupp."	Kan uppstå vid skapande/uppdaterande av artikel.
46	"Kan ej hitta angiven konteringskod."	Kan uppstå vid skapande/uppdaterande av artikel.
47	"Kan ej hitta angiven enhetskod."	Kan uppstå vid skapande/uppdaterande av artikel, order-, offert- eller fakturad.
48	"Kan ej hitta angiven leverantör."	Kan uppstå vid skapande/uppdaterande av artikel eller leverantörsfaktura.
49	"Kan ej hitta angiven kund."	Kan uppstå vid skapande/uppdaterande av en order, offert eller faktura.
50	"Kan ej hitta angiven artikel."	Kan uppstå vid skapande/uppdaterande av en order-, offert-, fakturad eller en leverantörsfakturad.
51	"Felaktigt kundnummer."	Används ej
75	"Databasen skapad med ett annat program."	Den databas som försöker öppnas är ej giltig för att användas med Visma Administration.
76	"Databasversionen är nyare än programmets databasversion."	Den databas som försöker öppnas är nyare än vad Visma Administration stöder.

77	"Databasversionen är äldre än programmets databasversion"	Den databas som försöker öppnas är äldre än vad Visma Administration stöder. Försök konvertera den med Visma Administration.
78	"Bof. Första posten passerad"	Kan uppstå när funktionen AdkFirst anropas.
79	"Eof. Sista posten passerad"	Kan uppstå när funktionen AdkNext anropas.
80	"Ingen Databas öppen."	Det finns ingen databas öppen. Använd AdkOpen för att öppna en databas.
102	"Okänt databasfält"	Används ej
103	"Felaktig datatyp"	Kan uppstå i AdkGet och AdkSet funktionerna när anropad funktion inte stämmer överens med fältets datatyp. T ex om AdkGetStr anropas med ett fält som är av typen eDouble.
104	"Felaktig sökordning"	Kan uppstå vid användandet av funktionen AdkSetSortOrder då angiven sökordning ej tillhör aktuell programdel.
105	"Ingen prislista angiven / Ingen normalprislista"	Om ingen normalprislista finns måste en prislista anges vid skapande/uppdaterande av artikel.
106	"Kan ej hitta leverantören i leverantörsregistret"	Används ej
107	"Felaktigt databasid"	Uppstår när den ADK_DATA som skickas in till en funktion har ett felaktigt databasid. Funktionen är ej möjligt på denna typ av ADK_DATA.
108	"Felaktig kontrollsiffra."	Ej tillåtet att ange felaktiga värden till fälten för plusgiro, bankgiro eller organisationsnummer.
109	"Ej tillåtet att ange värde till aktuellt fält"	Aktuellt fält är endast till för läsning. Kan även uppstå om fältet ej är möjligt att uppdatera för tillfället eller med önskat värde.
110	"Felaktig fakturatyp"	De tillåtna kundfakturatyperna är "F" för normalfaktura, "N" för kontantnota och "K" för kreditfaktura.
111	"Felaktigt index"	Angivet radnummer till order-, offertfaktura- eller leverantörsfakturarad är felaktigt.
112	"Blankt nyckelfält ej tillåtet"	Om inget orderradnummer är angivet vid arbete med kollihantering.
114	"För kort sträng. Måste vara 11 tecken"	Den sträng som skickas in till funktionen AdkLongToDate måste vara minst 11 tecken.
115	"Ingen träff på angivet sökvärde"	Sökvärde baseras på sökordning som anges med AdkSetSortOrder.
116	"Felaktig sökväg till företag"	Angiven sökväg till företag är felaktig.
118	"Ändring inte tillåten. Ingen prislista aktiv"	Kan uppstå vid arbete med en leverantör.
119	"Detta mottagarnummer används redan på en annan leverantör. Mottagarnumret måste vara unikt."	Mottagarnummer vid utlandsbetalningar måste vara unikt om leverantören är inställd på utlandsbetalningar.
120	"Felaktig avgiftskod"	Avgiftskoden för leverantörer vid utlandsbetalning måste vara korrekt. Korrekta värden är 0-3. 0 – avsändaren betalar allt 1 – mottagaren betalar allt 2 – mottagaren betalar expressavgiften 3 – mottagaren betalar utländska kostnader
121	"Ej tillåtet att radera språkkoden för svenska."	Används ej
122	"Ej tillåtet att radera språkkoden för det inhemska språket."	Används ej
123	"Ej tillåtet att radera språkkoden då den används för texter till dokumentmallarna"	Används ej
124	"Artikelnummer finns redan i listan"	Används ej
125	"Ej tillåtet att radera resultatenheten då den används på någon verifikation i år"	Används ej
126	"Radering ej tillåten, resultatenheten har ingående balans detta bokföringsår"	Används ej
127	"Radering ej tillåten, resultatenheten finns i budget detta bokföringsår"	Används ej

128	"Felaktigt nätlicensnummer: Nätlicensnumret och programversion stämmer ej"	Används ej längre
129	"Datorns klocka går fel."	Kontroll av datorns klockinställning.
130	"Din licens för Visma Administration är felaktig. Starta programmet, välj Om Visma Administration under Hjälp och klicka på Uppdatera licensen."	Kontroll av licensen sker i funktionen AdkOpen.
131	"Katalogen innehåller inte de delade filerna. Troligtvis en felaktig sökväg till programmet i filen spcsadm.ini. "	Sökvägen till programmet är fel, kan vara fel i filen spcsadm.ini.
132	" Versionen av API:et är äldre än versionen av Visma Administration. Uppdatera eller reparera installationen av Visma Administration."	Versionen av adk.dll är äldre än versionen av Visma Administration.
133	" Versionen av API:et är nyare än versionen av Visma Administration. Reparera installationen av Visma Administration."	Versionen av adk.dll är nyare än versionen av Visma Administration.
134	"Kan inte öppna filen "	Låsningsfilen ADM_LOCK.LCK gick ej att låsa.
135	"Du har inte tillräckligt med programlicenser. Maxantal är redan öppet"	API:et tar en licens från det antal licenser du har och det måste finnas en ledig då API:et skall startas.
136	"Ej tillåtet att uppdatera artikelnumret"	Ej tillåtet att uppdatera artikelnumret på en leverantörsfakturarad.
137	"Felaktig sökväg till gemensamma filer för Visma Administration."	Sökvägen till Visma Administration är felaktig.
139	"Inget fältnamn finns tillgängligt"	Det finns inget namn till aktuellt fält, kan uppstå vid anrop till funktionen AdkGetFieldName.
140	"Betalningen journalförd"	Används ej.
141	"Fakturan betald"	Ändring ej tillåten om fakturan är betald. Kan uppstå då en leverantörsfaktura skapas eller uppdateras.
142	"Det finns upplagda betalningar på fakturan."	Kan inte ändra flaggan för makulerad om det finns betalningar på leverantörsfakturan.
143	"Det finns rader knutna till den tidigare angivna leverantören."	Kan inte ändra leverantör på en leverantörsfaktura om det finns rader knutna till leverantören.
144	"Fakturaraden tillhör en typ som ej får ändras i detta fält."	Denna typ av leverantörsfakturarad får ej raderas.
145	"Fakturarader finns knutna till denna post."	Ej tillåtet att ändra kopplingsdokument på en leverantörsfakturarad om det finns rader knutna till kopplingsdokumentet.
146	"Du får inte skriva in en beställning som tillhör en annan leverantör än den som finns på fakturan."	Kan uppstå vid användandet av fältet för kopplingsdokument på en leverantörsfakturarad.
147	"Ej tillåtet att ändra fältet för samlingsfaktura på faktura"	Kan uppstå då en leverantörsfaktura skapas eller uppdateras.
148	"Kan ej hitta angivet konto."	Kan uppstå när ett skuldkonto är angivet på leverantörsfaktura eller när ett konto används på en order-, offert- faktura- eller leverantörsfakturarad.
149	"Kan ej hitta angivet projekt."	Kan uppstå när ett projekt är angivet på en order-, offert- faktura- eller leverantörsfaktura. Kan även uppstå när projekt används på order-, offert- faktura- eller leverantörsfakturarad.
150	"Kan ej hitta angiven resultatenhet."	Kan uppstå när en resultatenhet är angiven på en order, offert faktura eller en leverantörsfaktura. Kan även uppstå när resultatenhet används på en order-, offert-, faktura- eller en leverantörsfakturarad och angivet värde ej kan hittas.
151	"Inga artiklar på aktuell order går att leverera."	Kan uppstå vid försök till att sätta en order som levererad och inga artiklar går att leverera.
152	"Kan ej hitta angivet rabattavtal."	Kan uppstå när rabattavtal används på en order, offert faktura eller kund och angivet värde inte kan hittas.
153	"Felaktig kopplingstyp."	Används ej
155	"Kan inte radera posten då det finns dokument kopplade mot posten."	Kan ej radera en kund som har dokument kopplade till sig.
156	"Kan inte radera leverantören eftersom det finns en skuld till leverantören."	En leverantör är ej raderbar om den har en skuld.

157	"Kan ej radera artikel eftersom det finns artiklar i lagret."	Ej tillåtet att radera en artikel om den har positivt lagersaldo.
158	"Kan ej radera artikel eftersom det finns reserveringar."	Ej tillåtet att radera en artikel om det finns reservationer på den.
159	"Kan ej radera artikel eftersom det finns beställningar."	Ej tillåtet att radera en artikel om det finns beställningar på den.
160	"Kan ej radera artikel eftersom artikeln ingår i ett paket."	Ej tillåtet att radera en artikel om den ingår i en paketartikel.
161	"Kan ej radera eftersom fakturaraden tillhör en typ som ej får raderas."	Aktuell radtyp är ej raderbar
162	"Kan ej radera eftersom det finns fakturarader knutna till denna post."	Kan uppstå vid radering av leverantörsfakturarad.
163	"Kan ej radera eftersom endast planerade och direkta betalningar får raderas."	Kan uppstå vid radering av leverantörsfakturarad.
164	"Ej raderbar."	Ej tillåtet att radera denna typ av datastruktur.
166	"Kan ej hitta angiven beställning."	Den beställning som angivits som koppling på en leverantörsfakturarad kan ej hittas.
167	"Startdatum måste vara före slutdatum."	Vid skapande/uppdaterande av projekt måste startdatumet vara före slutdatumet.
168	"Projektet måste ha start- och slutdatum."	Vid skapande/uppdaterande av projekt måste startdatumet och slutdatumet finnas.
169	"Kan ej hitta angiven inkommande följesedel."	Kan ej läsa upp en inkommande följesedel på en leverantörsfaktura.
170	"Kopplingstyp och kopplingsdokument måste anges."	Anger du ett av fälten kopplingstyp eller kopplingsdokument på en leverantörsfakturarad så måste även det andra anges.
171	"Kan ej hitta angiven offert."	Används ej
172	"Kan ej hitta angiven order."	Ordern som angivits kan ej hittas.
173	"Kan ej hitta angiven faktura."	Används ej
174	"Kan ej hitta angivet avtal."	Används ej
175	"Kan ej hitta angiven journal."	Används ej
176	"Kan ej hitta angiven verifikation."	Används ej
177	"Inget sökvärde angivet."	Ett sökvärde på aktuellt fält måste anges vid sökning. Aktuellt fält bestäms av den sorteringsordning som är aktuell för tillfället.
180	"För många tecken"	För många tecken totalt eller före decimaltecknet i ett värde av typen <code>double</code> eller <code>date</code> .
181	"För många decimaler"	För många siffror efter decimaltecknet i ett värde av typen <code>double</code> .
182	"Inte tillåtet att uppdatera aktuellt fält på en textrad."	Fältet får ej tilldelas ett värde om radtypen i en order-, offert- eller fakturarad är satt till textrad
183	"Kontot finns inte, eller är inte märkt för projekt."	Kontot som angivits existerar ej.
184	"Projektnumret har använts tidigare, välj ett annat."	Projektnumret måste vara nytt och unikt.
185	"Företaget måste vara skapat med Visma Administration 2000."	Ej tillåtet att använda företag som ej är skapade i Administration 2000.
187	"Ingen databastabell kan härledas till detta fel."	Det fel som uppstått kan inte härledas till någon databastabell.
188	"Inget databasfält kan härledas till detta fel."	Det fel som uppstått kan inte härledas till något databasfält.
189	"Kvartalsrapportering kan inte anges om inget VAT-nummer finns."	Kan uppstå om en order, offert eller faktura ej har flaggan för export satt till sann.
190	"Du försöker använda en leverantörsföljesedel som är makulerad. Det får du inte göra."	Kan ej koppla en leverantörsfaktura till en följesedel som är makulerad.
191	"Du får inte skriva in en följesedel som tillhör en annan leverantör än den du har på fakturan."	Om en följesedel skall kopplas till en leverantörsfaktura måste leverantören vara den samma.
192	"Projekt är ej tillåtet att ange till aktuellt konto."	Kan uppstå när ett projekt försöker sättas på en leverantörsfaktura.
193	"Ej tillåtet att radera valutakoden för svenska."	Används ej
194	"Ej tillåtet att radera valutakoden för den inhemska valutan."	Används ej
195	"Inte tillåtet med textrad eller deltotal på rad med artikelnummer"	Kan ej omvandla en artikelrad till en textrad eller en deltotalrad.
196	"Inte tillåtet att uppdatera en textrad till en artikelrad"	Kan ej omvandla en textrad till en artikelrad.

197	"Radtypen är ej tillåten"	Tillåtna radtypen är "T"= textrad, "D" = deltotalrad och blankt för artikelrad.
198	"Inte tillåtet att uppdatera en textrad till en deltotalrad."	Ta bort raden och skapa en ny om raden önskas som deltotal.
199	"Ej tillåtet att uppdatera fakturatypen."	Fakturatypen kan ej uppdateras.
200	"Inte tillåtet att uppdatera en deltotalrad till en textrad."	Ta bort raden och skapa en ny om raden önskas som textrad.
201	"Inte tillåtet att uppdatera radtypen"	Den uppdatering av radtyp som önskas är ej tillåten.
202	"Kan inte ändra staffling då inget grundpris är angivet."	Ange ett grundpris, sedan är det möjligt att ändra stafflingen.
203	"Raden ingår i ett paket. Det är ej tillåtet att ändra leveransdatum på en enskild rad i ett paket."	Det är ej möjligt att uppdatera leveransdatum på en enskild rad i ett paket.
206	"Ändring inte tillåten. Om artikelnumret är fel får du radera raden och sedan skriva in nytt värde på en ny rad."	Det är inte tillåtet att uppdatera artikelnummer på en rad på en leverantörsfaktura.
207	"Företaget måste vara skapat med Visma Administration 1000."	Det företag som försöker öppnas är inte skapat med Visma Administration 1000.
208	"Fältet är inte tillgängligt i aktuell variant av programmet."	Aktuellt fält är inte tillgängligt i den variant av Visma Administration/Visma Förening som används.
209	"Hanteringen för språk och valuta är för tillfället inaktiverad."	Aktivera språk och valuta i Visma Administration.
210	"Hanteringen för streckkod är för tillfället inaktiverad."	Aktivera hanteringen av streckkod i Visma Administration.
211	"Ej tillåtet att utföra önskad operation på en NULL-pekare."	Den datastruktur som skickats in som en inparameter är en NULL-pekare.
212	"Ej tillåtet att utföra önskad operation på en NULL-pekare. Det är en radpekare som är NULL"	Den datastruktur som skickats in som en inparameter är en NULL-pekare. Det är radpekaren som är en NULL-pekare.
213	"Felaktig streckkodstyp"	Tillåtna streckkodstyper är 0-9. Se tabell i headerfil.
214	"Dokumentnumret finns redan."	Nästa nummer i nummerserien finns redan. Kontrollera i Visma Administration
215	"Artikelnumret finns inte angivet för aktuell leverantör."	Det artikelnummer som angivits finns inte angivet som "leverantörens artikelnummer" för aktuell leverantör.
216	"En ogiltig post försöker uppdateras."	Den post som försöker uppdateras är ogiltig, hämta en post med t ex AdkFind(), AdkFirst eller AdkLast().
217	"Ordern som angivits har redan genererat en faktura."	Den order som angivits som kopplingsdokument till offerten har genererat en faktura.
218	"Ordern som angivits har genererats från en annan offert."	Används ej.
219	"Offerten som angivits har redan genererat en annan order."	Den offert som angivits som kopplingsdokument till ordern har redan genererat en annan order.
220	"Ordern som angivits har redan genererat en annan faktura."	Används ej.
221	"Fakturan som angivits har genererats från en annan order."	Används ej.
222	"Kan ej hitta angiven streckkod."	Kan uppstå vid skapande/uppdaterande av en order-, offert-, fakturad eller en leverantörsfakturarad.
223	"Versionen av API:et stämmer inte överens med versionen av programmet."	Versionen på API:et och versionen på det program som pekas ut till AdkOpen () stämmer inte överens.
224	"Ändring inte tillåten på journalförd post."	Den aktuella manuella in eller utleveransen är journalförd och kan därför inte uppdateras.
225	"In och utleverans kan bara ske på artikel som är lagervara."	Angiven artikel måste vara en lagervara.
226	"Antal måste vara större än noll"	Angivet antal måste vara större än 0.
227	"Artikelnr måste finnas"	Ej tillåtet med blankt artikelnummer.
228	"Endast makulerade in och utleveranser kan raderas."	Makulera in eller utleveransen först och radera den sedan.
233	"Ej tillåtet att kreditera kreditfaktura."	Den faktura som ska krediteras kan ej vara av typen kredit.
234	"Ej tillåtet att kreditera en makulerad faktura."	Den faktura som ska krediteras kan ej vara makulerad.
235	"Felaktig kopieringstyp."	Felaktig kopieringstyp, se ADK_CREDIT_INVOICE_TO_COPY I adk.h.
236	"Kan ej hitta angiven speditör."	Kan uppstå vid skapande/uppdaterande av en order, offert, faktura eller kund.

237	"Ej tillåtet att använda aktuellt fält då fakturan inte är en kreditfaktura."	Fältet ADK_OOI_HEAD_CREDIT_NOTE_STORAGE_AFFECT kan endast användas på kreditfakturer.
261	"Kreditfaktura kan inte skapas från order."	Skapa faktura från order.
262	"Ordern finns inte."	Skapa faktura från order.
263	"Ordern makulerad."	Skapa faktura från order.
264	"Offerten finns inte."	Skapa order från offert.
265	"Offerten makulerad."	Skapa order från offert.
266	"Antal rader måste vara noll."	Skapa faktura från order och skapa order från offert.
267	"Ej tillåtet ändra kopplingsdokumentnummer."	Offert och faktura
268	"Du kan inte radera BIC eller IBAN-nummer när du markerat valet Betalning till annat EU-land i SEK/EUR."	Leverantörsregistret och leverantörsfakturer.
269	"Detta betalningsuppdrag kan endast väljas för betalningar i SEK eller EUR."	Leverantörsregistret och leverantörsfakturer.
270	"Felaktig BG-avgiftskod."	Leverantörsregistret och leverantörsfakturer.
271	"Vid valet Betalning till annat EU-land i SEK/EUR måste avgiftskoden vara Mottagaren betalar utländska kostnader."	Leverantörsregistret och leverantörsfakturer.
272	"Vald avgiftskod för utlandsbetalningar stöds inte av banken vald i företagsinställningarna."	Leverantörsregistret och leverantörsfakturer.
273	"Aktuellt fält får inte ändras i avslutat projekt."	Gäller alla fält utom flaggan för avslutat
274	"Företaget måste vara skapat med Visma Administration 500."	Ej tillåtet att använda företag som ej är skapade i Administration 500.
275	"Föreningen måste vara skapat med Visma Förening."	Ej tillåtet att använda föreningar som ej är skapade i Förening.
277	"Ändring inte tillåten. Faktura eller följesedel finns."	Beställningar.
278	"Databasen är inte tillgänglig i aktuell version av programmet."	Ej tillåtet att öppna register som inte finns i aktuell omfattning av Administration (Förening).
279	"Kan ej hitta angiven landskod."	Kunder, offerter, order, fakturer, beställningar och leverantörer.
282	"Du kan inte plocka bort flaggan Skriv på en rad med summa eftersom summan av fakturaderna då inte överensstämmer med fakturans totalbelopp."	Fakturer och order.
283	"Du kan inte lägga in en summa på en textrad som inte är märkt för skriv. Sätt i boken för skriv och försök igen."	Fakturer och order.
284	"Ändring inte tillåten. Dokumentet skickat med Autoinvoice."	Används ej.
286	"Huvudmedlem i familj kan ej raderas."	Medlemsregistret.
287	"Medlem i familj kan ej raderas."	Medlemsregistret.
288	"Markering för egen adress får inte tas bort på huvudmedlem i familj."	Medlemsregistret.
291	"Kan ej hitta angiven fri kategori 1."	Medlemsregistret.
292	"Kan ej hitta angiven fri kategori 2."	Medlemsregistret.
293	"Kan ej hitta angiven fri kategori 3."	Medlemsregistret.
294	"Kan ej hitta angiven fri kategori 4."	Medlemsregistret.
295	"Kan ej hitta angiven fri kategori 5."	Medlemsregistret.
296	"Kan ej hitta angiven fri kategori 6."	Medlemsregistret.
297	"Kan ej hitta angiven fri kategori 7."	Medlemsregistret.
298	"Kan ej hitta angiven fri kategori 8."	Medlemsregistret.
299	"Kan ej hitta angiven fri kategori 9."	Medlemsregistret.
300	"Kan ej hitta angiven fri kategori 10."	Medlemsregistret.
301	"Hanteringen av fri kategori 1 är inaktiverad."	Medlemsregistret.
302	"Hanteringen av fri kategori 2 är inaktiverad."	Medlemsregistret.
303	"Hanteringen av fri kategori 3 är inaktiverad."	Medlemsregistret.
304	"Hanteringen av fri kategori 4 är inaktiverad."	Medlemsregistret.
305	"Hanteringen av fri kategori 5 är inaktiverad."	Medlemsregistret.
306	"Hanteringen av fri kategori 6 är inaktiverad."	Medlemsregistret.
307	"Hanteringen av fri kategori 7 är inaktiverad."	Medlemsregistret.
308	"Hanteringen av fri kategori 8 är inaktiverad."	Medlemsregistret.
309	"Hanteringen av fri kategori 9 är inaktiverad."	Medlemsregistret.
310	"Hanteringen av fri kategori 10 är inaktiverad."	Medlemsregistret.
319	"Felaktigt angivet kön."	Medlemsregistret.
320	"Medlemmen är med i en familj och har inte egen adress markerat."	Medlemsregistret.
321	"Sökordningen är inte tillgänglig i aktuell version av programmet."	Sökordningen får inte användas i denna version av programmet (500/1000/ Förening)
322	"Ändring ej tillåten. Dokumentet markerat ej klart."	Beställning
323	"Ändring av namn på samlingsfaktura är inte tillåtet."	Faktura
324	"Ändring inte tillåten. Lagerföringsjournalförda rader finns på ordern."	Order
325	"Du måste ange BIC och IBAN-nummer."	Leverantör

329	"Medlemmen ingår i en familj. Ändring av adress/tel kan bara göras i medlemsregistret."	Kundregistret (endast i Visma Förening).
330	"Datum för periodstart kan endast anges för medlemsavgift/familjeavgift."	Faktura (Endast Visma Förening)
331	"Medlemsavgift/familjeavgift måste ha ett datum för periodstart."	Faktura (Endast Visma Förening)
333	"Kan inte byta leverantör. Beställning är upplöslad på följesedel."	Inkommande följesedel
334	"Kan inte byta artikel. Raden kommer från beställning."	Inkommande följesedel
335	"Ändring inte tillåten. Följesedeln är upptagen på en leverantörsfaktura."	Inkommande följesedel
336	"Du får inte skriva in en beställning som tillhör en annan leverantör än den du har på följesedeln."	Inkommande följesedel
337	"Radering inte tillåten. Följesedelsrader knutna till denna post finns."	Inkommande följesedel
338	"Konto för Skuld ink. Följesedlar är inte angivet."	Inkommande följesedel
339	"Ej tillåtet att uppdatera koppling, radera rad istället."	Inkommande följesedel
340	"Felaktig kopplingstyp"	Inkommande följesedel, Leverantörsfaktura
341	"Kan ej hitta angiven följesedelsrad."	Inkommande följesedel, Leverantörsfaktura
342	"Följesedelsraden kan ej kopplas till dokumentet."	Leverantörsfaktura
343	"Beställningsraden kan ej kopplas till dokumentet."	Inkommande följesedel, Leverantörsfaktura
344	"Ogiltig dokumenttyp."	Leverantörsfaktura
345	"En faktura med detta leverantörsnummer och fakturanummer finns redan."	Leverantörsfaktura
346	"Addering/Uppdatering av följesedelsrad är inkorrekt."	Inkommande följesedel, Leverantörsfaktura
347	"Beställningsraden kan ej kopplas till dokumentet. Obligatoriskt fält saknas."	Inkommande följesedel, Leverantörsfaktura
350	"Ej tillåtet att öppna återkopierat företag."	Kontrollerar att det ej är ett återkopierat företag
355	"Du kan inte fördela fler artiklar än vad du har kvar att leverera."	Kollihantering
356	"Du kan inte fördela ett negativt antal."	Kollihantering
357	"Ogiltigt referenskolli."	Kollihantering
358	"Ogiltigt orderrad."	Kollihantering
359	"Kolli kunde ej skapas."	Kollihantering
360	"Kan ej hitta angivet kolli."	Kollihantering
361	"Ändring inte tillåten. Leveransavisering skickad."	Kollihantering
362	"Du kan inte ändra en orderrad när det finns rader på ordern som ingår i ett eller flera kollin."	Kollihantering
363	"Kan ej hitta angiven förpackningstyp."	Kollihantering
364	"Webbshoppen är inte aktiverad."	Artikel
365	"EDI är inte aktiverat."	Leveransavisering
366	"Leveransavisering är inte skapad via Autoinvoice."	Används ej.
367	"Ändring av referenskolli nummer är inte tillåten."	Kollihantering
368	"Kollinummer kan ej anges vid skapande av kolli."	Kollihantering
369	"Ändring inte tillåten. Följesedel måste vara utskriven innan utskriftsval för leveransavisering kan väljas."	Order
370	"Ändring inte tillåten. Ordern makulerad."	Order
371	"Ändring inte tillåten. Ordern markerad som skickad."	Order
372	"Dokumentet har redan skickats och är under behandling. Det kan inte skickas på nytt innan det har fått en kvittens."	Order
373	"Momskod måste vara någon av siffrorna 0, 1, 2, 3 eller 4, eller mellanslag i vissa fall."	Offert, Order och Faktura
375	"Fakturan är skapad från en order som är markerad som levererad. Ändringen kan inte genomföras. Makulera denna faktura och gör ändringarna på ordern."	Faktura
376	"Summan får inte ändras eftersom textrad häftats till artikelrad."	Order
377	"Momsreglerna för omvänd skattskyldighet för vissa byggtjänster gäller inte vid export och försäljning till EU."	Inaktivt, mer information kommer om detta ska aktiveras
378	"Export och EU-försäljning omfattas inte av omvänd skattskyldighet för vissa byggtjänster."	Inaktivt, mer information kommer om detta ska aktiveras
379	"Fältet som används för omvänd skattskyldighet för vissa byggtjänster är inte aktiverat i företaget."	Kund, Offert, Order, Faktura och Artikelkontering
385	"SI-koden får endast vara kopplad till en enhet"	Enheter
387	"Artikels kontkod existerar inte"	Används av funktionen AdkGetVatpercentOnArticle
389	"Kan inte beräkna artikels pris för angiven kund"	Används av funktionen AdkGetCustomerPrice
390	"Artikeln saknar momscode"	Används av funktionen AdkGetVatpercentOnArticle
391	"Inhemsk nyckel måste skapas först"	Enhetskoder
392	"Otillåtet fält för utländsk kod försöker ändras eller skrivas"	Enhetskoder
393	"Otillåtet fält för inhemsk kod försöker ändras eller skrivas"	Enhetskoder

394	"Enheten får inte tas bort då det finns utländska benämningar"	Enhetskoder
397	"Decimaler måste ha något av följande värden 0, 1, 2, 3 eller 4"	Enhetskoder
399	"Grundpris måste anges först"	Kundinbetalningar
400	"Grundpriset får inte tas bort då det finns kopplade staffingpriser"	Kundinbetalningar
402	"Radtyp måste anges för raden"	Kundinbetalningar
403	"Radtyp måste ha något av följande värden H, A eller B"	Kundinbetalningar
404	"Betalsätt måste anges för inbetalningen"	Kundinbetalningar
405	"Bortskrivningskod måste anges"	Kundinbetalningar
406	"Ett konto måste anges för betalraden"	Kundinbetalningar
407	"Det angivna betalsättet är ogiltigt"	Kundinbetalningar
408	"Belopp måste vara angivet för betalraden"	Kundinbetalningar
409	"Både kurs och belopp i kr får inte anges"	Kundinbetalningar
410	"Fakturan måste vara markerad för skickad och får inte vara makulerad"	Kundinbetalningar
411	"Inget kontonummer angivet för Bankkostnader under Kontonummer under företagsinställningar"	Kundinbetalningar
412	"Inget kontonummer angivet för Kursdifferens under Kontonummer under företagsinställningar"	Kundinbetalningar
413	"Koppla radnummer måste anges"	Kundinbetalningar
414	"Avgiftspost finns redan kopplad till direktbetalningen"	Kundinbetalningar
415	"Bortskrivningspost finns redan kopplad till direktbetalningen"	Kundinbetalningar
416	"Det finns ingen obetald faktura med det fakturanumret"	Kundinbetalningar
417	"Betalsätt kan ej anges på denna radtyp"	Kundinbetalningar
418	"Bortskrivningskod kan ej anges på denna radtyp"	Kundinbetalningar
419	"Kopplad rad är ingen betalrad"	Kundinbetalningar
420	"Kopplad rad kan ej sättas på betalrad"	Kundinbetalningar
421	"Kan inte koppla till betalrad som är journalförd"	Kundinbetalningar
422	"Radtypen använder inte belopp i valuta, enbart belopp ska sättas"	Kundinbetalningar
423	"Den angivna bortskrivningskoden är ogiltig"	Kundinbetalningar
424	"Endast betalningsposter kan raderas direkt. Bortskrivningar, kursdiffar och avgifter raderas ihop med betalposten"	Kundinbetalningar
425	"Endast direktbetalningsposter kan tas bort"	Kundinbetalningar
426	"Journalförda betalposter kan ej tas bort"	Kundinbetalningar
427	"Radnummer kan inte sättas vid Add"	Kundinbetalningar
428	"Makulerade betalposter kan ej tas bort"	Kundinbetalningar
429	"Kvittnings mot annan faktura kan ej tas bort"	Kundinbetalningar
432	"Fältet får inte ha värdet 0.0"	Kundinbetalningar
433	"Betalningen har orsakat negativt saldo på fakturan"	Kundinbetalningar
435	"Fältet används inte längre"	Fält som slutat användas
436	"Kassakvitton måste ha markering för utskrivet"	Kassakvitto
437	"Markering får ej sättas manuellt"	Inköpspris
438	"Stafflingsantal måste vara större än noll"	Inköpspris
439	"Valuta måste ändras i leverantörsregistret"	Beställningar
440	"Ändring inte tillåten på textrad"	Inkommande följesedel
441	"Ändring inte tillåten. Artikeltransaktionen är journalförd"	Inkommande följesedel
442	"Felaktigt val av valuta"	Används av funktionen AdkGetCustomerPrice
443	"Momsreglerna för omvänd skattskyldighet för vissa byggtjänster gäller inte vid nedsatt egenavgift. Detta fält måste avmarkeras för att byggtjänst ska kunna anges."	Används inte från version 4.5
444	"Export och EU-försäljning omfattas inte av nedsatt egenavgift. Fältet för nedsatt egenavgift måste avmarkeras innan export eller EU-kund kan anges."	Används inte från version 4.5
445	"Nedsatt egenavgift gäller inte vid export och försäljning till EU. Dessa fält måste avmarkeras för att nedsatt egenavgift ska kunna anges."	Används inte från version 4.5
446	"Nedsatt egenavgift gäller inte vid omvänd skattskyldighet för vissa byggtjänster. Detta fält måste avmarkeras för att nedsatt egenavgift ska kunna anges."	Används inte från version 4.5
447	"Nedsatt egenavgift är inte aktiverat"	Används inte från version 4.5
448	"Du bör inte ta bort markeringen för skickad på fakturor som skickats eller hämtats med Visma Byråstöd – Kundfaktura."	Kundfakturer
449	"Faktura med markering för skattereduktion får inte ändras."	Kundfakturer
450	"Grundpriset får inte tas bort då det finns kopplade staffingpriser"	Pris
451	"Stafflingsantal måste vara större än noll"	Pris
452	"Procentsats går inte att sätta på grundpris"	Pris
453	"Grundpris måste anges först"	Pris
456	"Kan inte låsa post för ny verifikation"	Verifikationer
457	"Aktuellt år är låst"	Verifikationer
458	"Det finns ingen aktiv verserie med angiven beteckning"	Verifikationer
459	"Det finns inget att registrera"	Verifikationer
460	"Samtliga rader saknar belopp, det finns inget att registrera"	Verifikationer

461	"Månaden är låst"	Verifikationer
462	"Finns inget bokföringsår med angivet id"	Bokföringsår
463	"Konto för verifikationsraden är inte angivet"	Verifikationer
464	"Inget belopp är angivet på verifikationsraden"	Verifikationer
465	"Projekt måste anges på detta konto"	Verifikationer
466	"Resultatenhet måste anges på detta konto"	Verifikationer
467	"Det går inte att registrera en verifikation som inte balanserar"	Verifikationer
468	"Radinformation måste anges på detta konto"	Verifikationer
469	"Ej tillåtet med transaktionsinfo på detta konto"	Verifikationer
470	"Transaktionsdatumet ligger utanför verifikationens bokföringsår"	Verifikationer
471	"Ej tillåtet med subkonto på detta konto"	Verifikationer
472	"Subkonto måste anges på detta konto"	Verifikationer
473	"Kvantitet måste anges på detta konto"	Verifikationer
474	"Ej tillåtet med kvantitet på detta konto"	Verifikationer
477	"Ordern Ej klar"	Order
478	"Inga artiklar har levererat antal"	Order
483	"Kan inte skapa angiven målmap"	SIE
484	"Kan inte skapa SIE-fil"	SIE
485	"Organisationsnummret är felaktigt"	SIE
486	"Kan inte låsa företaget exklusivt"	Generellt
487	"Kombinering av artikelgrupp och artikel får ej göras, endast ett av fälten ska anges."	Rabattavtal
488	"Angivet fält-ID existerar inte för vald tabell."	Generellt
490	"Testdokument påverkar ditt lager och din bokföring på samma sätt som övriga dokument."	EDI
491	"Ogiltigt utskriftsval för denna dokumenttyp"	EDI
492	"Mottagaren vid val B-Brev via Autoinvoice måste vara i Sverige"	Används ej
493	"Information på kunden saknas för detta utskriftsval"	EDI
494	"En komplett adress för avsändaren saknas"	EDI
495	"En komplett adress till mottagaren saknas"	EDI
496	"Ogiltig e-postadress är angiven"	Generellt
498	"För att kunna skicka e-dokument måste du välja mottagare först."	EDI
499	"Dokumentet måste vara valt för att kunna ange utskriftsval"	EDI
502	"Samma titel får endast finnas upplagd en gång i registret"	Kontakttitlar
504	"Samma grupp får endast finnas upplagd en gång i registret"	Kontaktgrupper
506	"Kan ej hitta angiven kontakt"	Kontakter, kontaktkopplingar
507	"Kan ej hitta angiven kontaktgrupp"	Kontaktgrupper, kontaktkopplingar
508	"Kopplingen mellan kontakt och kontaktgrupp finns redan"	Kontaktkopplingar
511	"Ogiltigt telefonnummer"	Kontakter, Kunder, Leverantörer
513	"Namn måste finnas på en kontakt."	Kontakter
515	"Mobiltelefonnummer får ej anges då referens har valts."	Kunder, Leverantörer
516	"Kan ej hitta angiven signatur."	Kontakter
517	"Signatur får ej kombineras med referensnamn eller mobiltelefonnummer."	Kontakter
519	"Ogiltigt personnummer"	Skattereduktion
520	"Ansökan finns redan för personnumret på aktuell faktura."	Skattereduktion
521	"Ansökt belopp för fakturan är större än möjlig skattereduktion."	Skattereduktion
522	"Skattereduktion är inaktiverat."	Skattereduktion
523	"Fältet används inte av denna tabell."	Generellt
524	"Positiv skattereduktion måste anges."	Skattereduktion
526	"Rotarbete/grön teknik är inte aktiverat på detta dokument."	Faktura
527	"Skattereduktion är inte aktiverat på detta dokument."	Faktura
528	"Skattereduktion kan inte göras på dokument daterat före 2009-07-01."	Faktura
529	"Skattereduktion kan inte göras på dokument markerat med Export."	Faktura
530	"Skattereduktion kan inte göras på dokument markerat med Momsskyldig vid byggtjänster."	Faktura
531	"Skattereduktion kan inte göras på dokument markerat med EU Periodisk sammanställning."	Faktura
532	"Skattereduktion kan inte göras på dokument markerat med Mellanman i trepartshandel."	Faktura
533	"Skattereduktion kan inte göras på dokument med annan valuta än SEK."	Faktura
534	"Skattereduktion kan inte göras på dokument som är markerat för att skickas som e-faktura."	Faktura
536	"Ändring inte tillåten då dokumentet är markerat för skattereduktion."	Faktura
537	"Dokumentet kan inte skrivas ut. En eller flera poster under fördela skattereduktion saknar personnummer eller belopp."	Faktura
538	"Dokumentet kan inte skrivas ut. En eller flera poster under fördela skattereduktion överstiger möjlig skattereduktion per person."	Faktura

539	"Inte tillåtet att byta inhemsk valuta när Fakturerar husarbete/grön teknik är aktiverat i företaget."	Faktura
540	"Ändring inte tillåten. Någon annan användare skickar fakturor via Visma Byråstöd Online."	Faktura
541	"Ändring av valutainformation är inte tillåtet ett dokument som innehåller skattereduktion."	Faktura
542	"Kan inte markera arbetskostnad eller ändra Typ av skattereduktion på en rad vars dokument inte har markeringen för skattereduktion."	Faktura
543	"Endast normala fakturor kan innehålla skattereduktion."	Faktura
544	"Rader i paket och strukturartiklar kan inte markeras som arbetskostnad."	Faktura
545	"Du får inte markera ränterad eller kravrad som arbetskostnad."	Faktura
546	"Angivet belopp är större än maximal reduktion per person."	Skattereduktion
548	"Radering av betalning inte tillåten. Betalningen är kopplad till ansökan eller beslut om skattereduktion."	Kundinbetalningar
549	"Du försöker använda en leverantörsföljesedel som är fakturerad. Det får du inte göra."	Leverantörsfaktura
550	"<Alla> är reserverad för internt bruk."	Kontakttitlar, kontaktgrupper
551	"Fastighets-/lägenhetsbeteckning får inte vara tom vid rotarbete/grön teknik."	Skattereduktion
553	"Projekt är avslutat och du kan därför inte registrera på detta projekt."	Avtal
555	"Avtalets startdatum måste ligga före slutdatumet."	Avtal
556	"Avtalets slutdatum måste ligga efter startdatumet."	Avtal
557	"Avtalets startdatum får inte lämnas blankt."	Avtal
558	"Periodens startdatum måste ligga före slutdatumet."	Avtal
559	"Periodens slutdatum måste ligga efter startdatumet."	Avtal
560	"Journalföring pågår."	Generellt
561	"Fältet på avtalet får inte ändras efter att fakturor har skapats."	Avtal
562	"Faktura skapad från avtalet. Ändringen gäller för kommande fakturor."	Avtal
563	"Fakturamånaden måste ligga i intervallet -2 till 12."	Avtal
564	"Fakturadagen måste ligga i intervallet -99 till 31."	Avtal
565	"Fakturaintervallet måste vara större än 0."	Avtal
568	"Endast makulerade eller avslutade avtal kan raderas."	Avtal
569	"Ändring inte tillåten. Avtalet är avslutat."	Avtal
570	"Ändring av valutainformation är inte tillåtet på en faktura som innehåller ränterader eller kravavgiftsrader."	Faktura
571	"Avtalet är makulerat."	Faktura
572	"Avtalet är avslutat."	Faktura
573	"Faktura kan inte skapas från en kombination av dokument."	Faktura
574	"Ogiltigt val av datumkod."	Faktura
575	"Datumangivelse saknas."	Faktura
576	"Avtalet saknas."	Faktura
578	"Avtalet har löpt ut. Avsluta eller förläng avtalet."	Avtal
580	"Du måste aktivera Visma Administration för att överföringen ska fungera. Starta programmet och välj Aktivera."	AdkOpen
581	"Du har en nyare version av Visma Administration än du har licens för. Starta programmet, välj Om Visma Administration under Hjälpen och klicka på Uppdatera licensen."	AdkOpen
582	"Ditt serviceavtal för Visma Administration gäller inte längre. Starta programmet, välj Om Visma Administration under Hjälpen och klicka på Uppdatera licensen."	AdkOpen
583	"Din licens för Visma Administration gäller inte för Visma Integration. Kontakta Visma Spcs för information och ändring av din licens."	AdkOpen
586	"För kort sträng. Måste vara 19 tecken"	AdkLongToDateTime
587	"Felaktigt format på datum/tid."	AdkLongToDateTime
591	"Förfallodatum måste anges med uppgifter om år, månad och dag."	Faktura
592	"Du kan inte ändra landskod eftersom du valt utskriftsval EDI/Brev B. Detta utskriftsval kan endast kombineras med landskoden SE (Sverige)."	Faktura
593	"Du kan inte plocka bort flaggan Skriv följesedel på en artikelrad eftersom antalet artiklar då inte överensstämmer mellan följesedel och leveransavisering."	Används ej
594	"En annan faktura är skapad för den inkommande följesedeln. Det går inte att upphäva makuleringen."	Leverantörsfaktura
595	"Det finns transaktioner på kontot under aktivt bokföringsår."	Konto
596	"Kontot har ingående balans detta bokföringsår."	Konto
597	"Kontot har saldo detta bokföringsår."	Konto
598	"Kontot finns i budget detta bokföringsår."	Konto
599	"Kontot finns i projekt."	Konto
601	"Angivet datum finns inte inom något bokföringsår."	Generellt

602	"Du har inget serviceavtal för Visma Administration, vilket krävs för att integrationen ska fungera."	AdkOpen
607	"Fält för tidstämpel saknas i databasen"	AdkGetTimestampField
608	"Du kan inte ange både signatur i Ref_Sign och namn i Referens på samma post, då fälten är beroende av varandra."	Offert, order, faktura, avtal
609	"Du kan inte ange något i fältet RefXXX_Id när du för dokumenttypen har valt E eller blankt på RefXXX_Typ. XXX är dokumenttypen."	Kund
610	"Du måste ange signatur för kundens kontakt i fältet RefXXX_Id för dokumentet när du har K för RefXXX_Typ. XXX är dokumenttypen."	Kund
611	"Du kan inte ange K för fältet RefXXX_Typ på en ny kund. XXX är dokumenttypen."	Kund
612	"Det är inte tillåtet med några andra tecken än K, E eller blankt på RefXXX_Typ. XXX är dokumenttypen."	Kund
613	"Det är inte tillåtet med några andra tecken än K eller blankt på RefKrv_Typ."	Kund
615	"Leverantörsfakturan är kopplad till en periodisering och får därför inte ändras/raderas."	Leverantörsfaktura
619	"Artikeln är en webbshopsartikel och kan därför inte markeras som inaktiv."	Artikel
620	"Artikeln är markerad som inaktiv och kan därför inte markeras som webbshopsartikel."	Artikel
621	"Summan av skattereduktionen överstiger belopp att betala."	Rotavdrag
622	"Konto måste anges."	Generellt
623	"Ändring inte tillåten. Lagerförändringsjournalförda rader finns på fakturan."	Faktura
624	"Du kan inte ändra utskriftsval när dokumentet är utskrivet."	Faktura
627	"Startdatum saknas"	Periodisering
628	"Antal månader måste vara mellan 1 och 99."	Periodisering
629	"Periodisering finns redan för raden."	Periodisering
630	"Kan inte hitta dokumentet eller raden som ska periodiseras."	Periodisering
631	"Kan inte skapa periodisering."	Periodisering
632	"Konto saknas på kopplad rad."	Periodisering
633	"Belopp saknas på kopplad rad."	Periodisering
634	"Kredit- och räntefakturor kan inte periodiseras."	Periodisering
635	"Periodisering kan inte skapas på faktura som saknar datum."	Periodisering
636	"Periodisering kan inte skapas för fakturans totalrad."	Periodisering
637	"Periodisering kan inte skapas för fakturans artikelrader."	Periodisering
638	"Verifikationsserie måste anges."	Periodisering
639	"Verifikationsserien är inställd som serie för redovisningsbyrå."	Periodisering
640	"Ändring inte tillåten. Fakturan är en kreditfaktura."	Leverantörsfaktura
641	"Ändring inte tillåten. Fakturan är en räntefaktura."	Leverantörsfaktura
642	"Ändring inte tillåten. Fakturan är märkt för autogiro."	Leverantörsfaktura
643	"Ändring inte tillåten. Fakturan är spärrad för betalning."	Leverantörsfaktura
644	"Ändring inte tillåten. Fakturan är redan slutbetald."	Leverantörsfaktura
645	"Ändring inte tillåten. Fakturan har ett negativt saldo."	Leverantörsfaktura
646	"Otillåtet tecken. Prislistans benämning får inte innehålla mellanslag, komma eller semikolon."	Prislista
647	"Prislistan är en webbshopsprislista och du kan därför inte ändra valuta."	Prislista
650	"Alla månaderna i bokföringsåret är låsta. Registrering är endast tillåten på sista dagen i bokföringsåret."	Verifikation
652	"Arbetskostnad kan inte ändras på en rad med Typ av skattereduktion angiven."	Avtal, Order, Faktura
653	"Typ av Rot/Rut strider mot inställning för skattereduktion på dokumentet."	Avtal, Order, Faktura
654	"Ogiltigt värde för Typ av skattereduktion."	Artikel, Avtal, Order, Faktura
655	"Arbetade timmar kan bara anges på rad med arbete."	Faktura
656	"Ändring inte tillåten. Ansökan om skattereduktion är redan gjord i programdelen Skattereduktion."	Faktura
657	"Fakturan är markerad som skickad. Du kan inte byta till material eller övrigt på rad som är arbete."	Faktura
658	"Fakturan är markerad som skickad. Du kan inte byta till arbete på rad som är material eller övrigt."	Faktura
659	"Radering av sista kvarvarande betalning inte tillåten. Fakturan är markerad för Delansökt om skattereduktion."	Kundinbetalning
660	"Negativt värde inte tillåtet för arbetade timmar."	Faktura
661	"Betalning kan inte skapas. Fakturan är överlämnad till AutoCollect."	Kundinbetalning
662	"Ej tillåtet att kreditera en faktura som är överlämnad till AutoCollect. För att kreditera fakturan måste den först återkallas från AutoCollect."	Faktura
663	"Ändring inte tillåten. Fakturan är överlämnad till AutoCollect."	Faktura

664	"Radering av betalning inte tillåten. Fakturan är överlämnad till AutoCollect."	Kundinbetalning
666	"Fakturan är kopplad till en periodisering och får därför inte ändras/raderas."	Faktura
671	"Nullpekare använd."	AdkGetDecimals, AdkGetLength, AdkGetType, AdkGetLastFieldId, AdkGetNrowsFieldId, AdkGetRowsFieldId, AdkGetRowDbId, AdkIsLoginActivated
672	"Ogiltig datastruktur."	Funktioner som tar en PADK_DATA (ADK_DATA*) som inparameter
673	"Databasen behöver uppgraderas."	AdkOpen, AdkOpen2, AdkOpenEx, AdkOpenEx2
674	"Inloggning nekades."	AdkOpen, AdkOpen2, AdkOpenEx, AdkOpenEx2
675	"Licensen stödjer inte denna typ av SQL server."	AdkOpen, AdkOpen2, AdkOpenEx, AdkOpenEx2
676	"Inloggning misslyckades."	AdkOpen, AdkOpen2, AdkOpenEx, AdkOpenEx2
685	"Ogiltigt fält att sätta filter på."	AdkSetFilter
686	"Ogiltigt filteruttryck."	AdkSetFilter
687	"Ogiltig version för filtersyntax."	AdkSetFilter
690	"Ändring inte tillåten på faktura med skattereduktion inläst från Visma eEkonomi Fakturering."	Faktura
691	"Ändring av låsta fält inte tillåten."	Artikel, Levvillkor, Levsätt, Detsätt, Artikelkontering, Enhetskod, Resenhet, Kund, Faktura, Projekt
692	"Ändring inte tillåten då dokumentet är markerat för undantag från skatteplikt."	Faktura
693	"Dokumentet kan inte markeras för både husarbete och grön teknik."	Faktura, Order
694	"Husarbete är inte aktiverat på detta dokument."	Faktura, Order
695	"Fakturan kan inte ändras eftersom den är såld via Visma Betalt Direkt."	Faktura

Tabeller

Kod	Felmeddelande	Förklaring
52	"Kund."	Felet har uppstått i databastabellen för kunder.
53	"Art."	Felet har uppstått i databastabellen för artiklar.
54	"Oof."	Felet har uppstått i databastabellen för order, offert och faktura.
55	"ArtRad."	Felet har uppstått i databastabellen för Artikelrad.
56	"Kundka."	Felet har uppstått i databastabellen för kundkategorier.
57	"Distri."	Felet har uppstått i databastabellen för distrikt.
58	"Saelja."	Felet har uppstått i databastabellen för säljare.
59	"Levvil."	Felet har uppstått i databastabellen för leverantörsvillkor.
60	"Betvil."	Felet har uppstått i databastabellen för betalningsvillkor.
61	"Valutk."	Felet har uppstått i databastabellen för valutakoder.
62	"Sprak."	Felet har uppstått i databastabellen för språk.
63	"Prisl."	Felet har uppstått i databastabellen för prislister.
64	"Artgrp."	Felet har uppstått i databastabellen för artikelgrupper.
65	"Artknt."	Felet har uppstått i databastabellen för artikelkontering.
66	"Enhkod."	Felet har uppstått i databastabellen för enhetskoder.
67	"Lev."	Felet har uppstått i databastabellen för leverantörer.
68	"Prisl."	Felet har uppstått i databastabellen för prislister.

69	"Levfkt."	Felet har uppstått i databastabellen för leverantörsfakturor.
70	"Konto."	Felet har uppstått i databastabellen för kontokoder.
71	"Proj."	Felet har uppstått i databastabellen för projekt.
72	"levsae"	Felet har uppstått i databastabellen för leveranssätt.
73	"Kundrbt."	Felet har uppstått i databastabellen för kundrabatter.
74	"Resenh."	Felet har uppstått i databastabellen för resultatenheter.
229	"Art. – Inventering."	Felet har uppstått i databastabellen för inventering.
230	"ArtRad. – Manuell inleverans."	Felet har uppstått i databastabellen för manuella inleveranser.
231	"ArtRad. – Manuell utleverans."	Felet har uppstått i databastabellen för manuella utleveranser.
232	"Speditor."	Felet har uppstått i databastabellen för speditör.
276	"Bst."	Felet har uppstått i databastabellen för beställningar.
280	"Pris."	Felet har uppstått i databastabellen för försäljningspriser.
285	"Medlemmar (kund)"	Felet har uppstått i databastabellen för medlemmar.
311	"Artpaket."	Felet har uppstått i databastabellen för paketartiklar.
312	"Inkpris."	Felet har uppstått i databastabellen för inköpspriser.
313	"Order (oof)"	Felet har uppstått i databastabellen för order.
314	"Orderrad (artrad)"	Felet har uppstått i databastabellen för orderrader.
315	"Offert (oof)"	Felet har uppstått i databastabellen för offerter.
316	„Offertrad (artrad)“	Felet har uppstått i databastabellen för offertrader.
317	"Beställningsrad (artrad)"	Felet har uppstått i databastabellen för beställningsrader.
318	"Fria kategorier (frikat)"	Felet har uppstått i databastabellen för fria kategorier.
332	"InkFlj."	Felet har uppstått i databastabellen för inkommande följesedlar
351	"Kolli."	Felet har uppstått i databastabellen för kollihantering
352	"Kollirad."	Felet har uppstått i databastabellen för Kollihantering
353	"LevaviKolli."	Felet har uppstått i databastabellen för Kollihantering
354	"LevaviKollirad."	Felet har uppstått i databastabellen för Kollihantering
374	"AvvikLevAdr."	Felet har uppstått i databastabellen för Avvikande leveransadress
395	"ErReferensKund (errefkund)"	Felet har uppstått i databastabellen för Er Referens, Kund
396	"ErReferensLeverantör (errevlev)"	Felet har uppstått i databastabellen för Er Referens, Leverantör
398	"Land."	Felet har uppstått i databastabellen för landskoder
401	"KbetBnt2."	Felet har uppstått i databastabellen för kundinbetalningar
430	"Betalsätt (Betsaett)"	Felet har uppstått i databastabellen för Betalsätt
431	"Bortskrivningar (Kbetbort)"	Felet har uppstått i databastabellen för Bortskrivningar
434	"Utbetalningar (Levbetspec)"	Felet har uppstått i databastabellen för Utbetalningar
454	"Ver."	Felet har uppstått i databastabellen för Verifikationer
455	"Verrad."	Felet har uppstått i databastabellen för Verifikationsrader
475	"Bokfaar."	Felet har uppstått i databastabellen för Bokföringsår

481	"RABKFILT."	Felet har uppstått i databastabellen för Rabattkoder
489	"Prm."	Felet har uppstått i databastabellen för Företagsinställningar
500	"Kontakt."	Felet har uppstått i databastabellen för Kontakter
501	"KontaktTit."	Felet har uppstått i databastabellen för Kontakttitlar
503	"KontaktGrp."	Felet har uppstått i databastabellen för Kontaktgrupper
505	"KontaktKop."	Felet har uppstått i databastabellen för Kontaktkopplingar
518	"SkvHus."	Felet har uppstått i databastabellen för Skattereduktion
554	"Avtal (Oof)"	Felet har uppstått i databastabellen för Avtal
588	"Kundrbt2."	Felet har uppstått i databastabellen för Kundrabattrader
589	"Artben."	Felet har uppstått i databastabellen för Artikelbenämningar
590	"Prm2."	Felet har uppstått i databastabellen för Företagsinställningar2
603	"Skattereduktion order (Skvhus)"	Felet har uppstått i databastabellen för Skattereduktion på order
604	"Skattereduktion husarbete avtal (Skvhus)"	Felet har uppstått i databastabellen för Skattereduktion på avtal
605	"Verserie."	Felet har uppstått i databastabellen för Verifikationsserier
614	"Bokhist."	Felet har uppstått i databastabellen för Bokföringshistorik
618	"Vermall."	Felet har uppstått i tabellen för Periodiseringar
648	"KundArt."	Felet har uppstått i tabellen för Kundartiklar
649	"Kopplund."	Felet har uppstått i tabellen för Kopplingsunderlag
651	"HusArbTyper."	Felet har uppstått i tabellen för Husarbetstyper
665	"Prm3."	Felet har uppstått i databastabellen för Företagsinställningar3

Funktioner

Kod	Felmeddelande	Förklaring
81	"AdkAdd"	Felet har uppstått i funktionen AdkAdd.
82	"AdkFind"	Felet har uppstått i funktionen AdkFind.
83	"AdkFirst"	Felet har uppstått i funktionen AdkFirst.
84	"AdkNext"	Felet har uppstått i funktionen AdkNext.
85	"AdkUpdate"	Felet har uppstått i funktionen AdkUpdate.
86	"AdkSetLong"	Felet har uppstått i funktionen AdkSetLong.
87	"AdkSetBool"	Felet har uppstått i funktionen AdkSetBool.
88	"AdkSetDouble"	Felet har uppstått i funktionen AdkSetDouble.
89	"AdkSetStr"	Felet har uppstått i funktionen AdkSetStr.
90	"AdkSetData"	Felet har uppstått i funktionen AdkSetData.
91	"AdkSetDate"	Felet har uppstått i funktionen AdkSetDate.
92	"AdkGetStr"	Felet har uppstått i funktionen AdkGetStr.
93	"AdkGetLong"	Felet har uppstått i funktionen AdkGetLong.
94	"AdkGetBool"	Felet har uppstått i funktionen AdkGetBool.
95	"AdkGetDouble"	Felet har uppstått i funktionen AdkGetDouble.
96	"AdkGetDate"	Felet har uppstått i funktionen AdkGetDate.

97	"AdkGetType"	Felet har uppstått i funktionen AdkGetType.
98	"AdkGetData"	Felet har uppstått i funktionen AdkGetData.
99	"AdkGetDataRow"	Felet har uppstått i funktionen AdkGetDataRow.
100	"AdkOpen"	Felet har uppstått i funktionen AdkOpen.
101	" AdkDeleteStruct "	Felet har uppstått i funktionen AdkDeleteStruct.
113	"AdkLongToDate"	Felet har uppstått i funktionen AdkLongToDate.
117	"AdkSetSortOrder"	Felet har uppstått i funktionen AdkSetSortOrder.
138	"AdkDateToLong"	Felet har uppstått i funktionen AdkDateToLong.
154	"AdkDelete "	Felet har uppstått i funktionen AdkDelete.
178	"AdkGetLength"	Felet har uppstått i funktionen AdkGetLength.
179	"AdkGetDecimals"	Felet har uppstått i funktionen AdkGetDecimals.
165	"AdkDeleteRow"	Felet har uppstått i funktionen AdkDeleteRow.
186	"AdkGetFieldName"	Felet har uppstått i funktionen AdkGetFieldName.
204	"AdkPrevious"	Felet har uppstått i funktionen AdkPrevious.
205	"AdkLast"	Felet har uppstått i funktionen AdkLast.
281	"AdkGetAdmSize"	Felet har uppstått i funktionen AdkGetAdmSize.
348	"AdkOpenEx"	Felet har uppstått i funktionen AdkOpenEx.
349	"ResetStruct"	Felet har uppstått i funktionen ResetStruct.
380	"AdkFirstEx"	Felet har uppstått i funktionen AdkFirstEx
381	"AdkNextEx"	Felet har uppstått i funktionen AdkNextEx
382	"AdkFindEx"	Felet har uppstått i funktionen AdkFindEx
383	"AdkPreviousEx"	Felet har uppstått i funktionen AdkPreviousEx
384	"AdkLastEx"	Felet har uppstått i funktionen AdkLastEx
386	"AdkGetVatpercentOnArticle"	Felet har uppstått i funktionen AdkGetVatpercentOnArticle
388	"AdkGetCustomerPrice"	Felet har uppstått i funktionen AdkGetCustomerPrice
476	"AdkGetVariant"	Felet har uppstått i funktionen AdkGetVariant
482	"AdkGetSie"	Felet har uppstått i funktionen AdkGetSie
577	"AdkFind2"	Felet har uppstått i funktionen AdkFind2
584	"AdkLongToDateTime"	Felet har uppstått i funktionen AdkLongToDateTime
585	"AdkDateTimeToLong"	Felet har uppstått i funktionen AdkDateTimeToLong
600	"AdkGetBookingYearId"	Felet har uppstått i funktionen AdkGetBookingYearId
606	"AdkGetTimeStampField"	Felet har uppstått i funktionen AdkGetTimeStampField
616	"AdkGetCompanyName"	Felet har uppstått i funktionen AdkGetCompanyName
617	"AdkGetCompanyNameEx"	Felet har uppstått i funktionen AdkGetCompanyNameEx
625	"AdkGetCustomerPrice2"	Felet har uppstått i funktionen AdkGetCustomerPrice2
626	"AdkGetCompanySetting"	Felet har uppstått i funktionen AdkGetCompanySetting
667	"AdkGetLastFldId"	Felet har uppstått i funktionen AdkGetLastFieldId
668	"AdkGetNrowsFldId"	Felet har uppstått i funktionen AdkGetNrowsFieldId

669	"AdkGetRowsFldId"	Felet har uppstått i funktionen <code>AdkGetRowsFieldId</code>
670	"AdkGetRowDbId"	Felet har uppstått i funktionen <code>AdkGetRowDbId</code>
677	"AdkOpen2"	Felet har uppstått i funktionen <code>AdkOpen2</code>
678	"AdkOpenEx2"	Felet har uppstått i funktionen <code>AdkOpenEx2</code>
679	"AdkIsLoginActivated"	Felet har uppstått i funktionen <code>AdkIsLoginActivated</code>
680	"AdkAddEx"	Felet har uppstått i funktionen <code>AdkAddEx</code>
681	"AdkUpdateEx"	Felet har uppstått i funktionen <code>AdkUpdateEx</code>
682	"AdkSetFilter"	Felet har uppstått i funktionen <code>AdkSetFilter</code>
683	"AdkIsFilterable"	Felet har uppstått i funktionen <code>AdkIsFilterable</code>
684	"AdkClearFilters"	Felet har uppstått i funktionen <code>AdkClearFilters</code>

Varningar (IProgramPart: APP (=4))

Kod	Varning	Förklaring
238	"Kreditgränsen överskriden på angiven kund"	Uppstår om en order skapas till en kund som överskridit kreditgränsen.
239	"Mottagnummer är bankens unika nyckel för våra betalningar till leverantören. Den ska normalt sett aldrig ändras, detta har dock gjorts nu"	Kan uppstå vid arbete med leverantörers utlandsbetalningar.
240	"BIC får enbart innehålla siffror, bokstäver eller frågetecken. Mellanslag är ej tillåtna. Fältet har justeras automatiskt."	Kan uppstå vid arbete med leverantörers utlandsbetalningar.
241	"Ändring av valutaenhet kommer att påverka alla befintliga dokument som har använt aktuell valuta."	Används ej.
242	"OCR/Referensnummer används ej på utländska fakturor."	Ej tillåtet att ange ett OCR/Referensnummer på leverantörsfaktura.
243	"Leverantören är inställd att inte använda OCR/ Referensnummer."	Aktuell leverantör på leverantörsfakturan är ej inställd att använda OCR/Referensnummer.
244	"Angivet plusgironr stöder inte OCR/Referensnrbetalningar."	Kan uppstå vid arbete med leverantörsfakturer.
245	"Du försöker ändra på ett fält i en rad som kommer från en följesedel. Om du godkänner ändringen kommer även följesedelsraden att ändras."	Kan uppstå vid arbete med leverantörsfakturarader.
246	"Artikeln har lagertransaktioner. Radering bör undvikas."	Kan uppstå vid arbete med artiklar.
247	"Radera ordersumma. Om du raderar en ordersumma kan du inte återskapa den."	Kan uppstå vid försök att radera en faktura om den är satt som samlingsfaktura.
248	"Verifikationer finns på projektet."	Kan uppstå vid försök att radera projekt.
249	"Denna leverantör har inte markering för Kräver OCR/Refnr men plusgironr är ett OCR-nr."	Leverantörens plusgironummer motsvarar ett OCR-nummer.
250	"Denna leverantör har markering för Kräver OCR/Refnr men plusgironr är inget OCR-nr."	Leverantörens plusgironummer motsvarar inte ett OCR-nummer.
251	"Kan ej hitta pris till artikeln i aktuell prislista(normalprislistan om ingen prislista är angiven)."	Pris på artikel måste existera när den hämtas med <code>AdkFirst</code> , <code>AdkNext</code> eller <code>AdkFind</code> .
252	"Det finns artikeltransaktioner för denna artikel. Det finns risk för att framtida lagerförändringsjournaler, lagervärden mm kan bli felaktiga."	Artikeln som raderas har artikeltransaktioner.
253	"Kan ej finna konteringsmallen som är kopplad till leverantören. Standardmallen kommer att användas istället."	Den angivna konteringsmallen kan inte hittas.
254	"Konteringsmallen som är kopplad till leverantören är ej avsedd för att användas i leverantörsfakturaregistreringen. Standardmallen kommer att användas istället."	Den angivna konteringsmallen är inte avsedd att användas i leverantörsfakturaregistreringen.
255	"Raden som uppdateras är en rubrikrad eller en låst rad. Raden kommer att behålla sin status som rubrikrad eller låst rad."	Om raden som uppdateras är en rubrikrad eller en låst rad.
256	"Angivet konto är inte aktivt."	Det konto som angivits är inaktivt.
257	"Inventerad artikel saknar kalkylpris."	Artikeln som inventeras saknar kalkylpris
258	"Den manuella inleveransen har inget pris/st. Detta innebär att inleveransen ej kommer att påverka lagervärdet."	Lagervärdet har inte uppdaterats då pris saknas.
259	"Fakturarader som tillhörde följesedlar kopierades inte."	När en kreditfaktura skapas utan valet "kopiera allt".
260	"Denna varningskod används inte längre. (Fr.o.m version 4.1)"	Används ej längre.
289	"Misslyckades ladda adress från huvudmedlem i familj."	Medlemsregistret.

290	"Misslyckades ladda adress till medlem i familj."	Medlemsregistret.
326	"Byte av valuta efter att rader lagts in på dokumentet. För att raderna ska uppdateras med den nya valutan, måste dessa läggas in på nytt."	Offert, order och faktura.
327	"Byte av valutakurs efter att rader lagts in på dokumentet. För att raderna ska uppdateras med den nya valutan, måste dessa väljas in på nytt."	Offert, order och faktura
328	"Byte av kund, med annan valuta än den som fanns när artikelraderna lades in på dokumentet. För att raderna ska uppdateras med den nya valutan, måste dessa läggas in på nytt."	Offert, order och faktura
448	"Du bör inte ta bort markeringen Utskrivet på fakturor som importerats."	Markeringen bör ej tas bort då fakturan därefter kan ändras, vilket innebär att den inte är identisk med orginfakturan.
479	"Kontrollsiffran är felaktig på EAN-Lokaliseringnr"	Kund- och Leverantörsuppgifter
480	"Leverantören är ej markerad som utländsk. Valutakonton används enbart på leverantörsbetalningar i utländsk valuta."	Leverantör
497	"Betalningsvillkor bör vara angivet."	Används ej
509	"En signatur måste finnas på kontakten"	Kontakter
510	"Signaturen finns redan för kopplad kund eller leverantör alternativt okopplade kontakter. Kontrollera genererad signatur."	Kontakter
512	"Adressuppgifterna stämmer inte överrens, var god kontrollera adressuppgifterna."	Kontakter
514	"Ett mobiltelefonnummer bör innehålla minst 9 siffror och max 14 siffror."	Kontakter
525	"Bostadsrättsföreningens organisationsnummer är inte giltigt."	Faktura
535	"Då kunden byts på dokumentet har fördelningen av skattereduktion nollställts, fördelningarna måste läggas upp på nytt."	Faktura
547	"Summan av skattereduktionen understiger möjlig skattereduktion."	Skattereduktion
547	"Summan av skattereduktionen understiger möjlig skattereduktion."	Skattereduktion
552	"Ett borttaget fält försöker användas."	Generellt
566	"På de artikelrader som berörs av momsskyldig vid byggtjänst måste du manuellt ändra försäljningskonton och momskod."	Avtal
567	"När EU- eller exportflaggan ändras kommer artikelradernas försäljningskonton inte att ändras automatiskt."	Offert, order, faktura, avtal
579	"När EU- eller exportflaggan ändras kommer artikelradernas försäljningskonton inte att ändras automatiskt."	Order, faktura, avtal
689	"Du bör inte ta bort markeringen för skickad på fakturor som är inlästa från extern källa."	Faktura

Grundstruktur (IProgramPart: FW (=2))

1000	"Ok"	Inget fel har uppstått.
1001	"Internt fel, kontakta SPCS"	Allvarligt fel.
1002	"Redigering ej tillåten"	Önskad radering kan ej utföras.
1003	"Redigering ej tillåten, ingen post är aktiv"	Önskad radering kan ej utföras.
1004	"Ej tillåtet att redigera nyckelfältet"	Önskad radering kan ej utföras.
1005	"Misslyckades med att låsa posten"	Postlåsningsfel, kontrollera om posten är aktiv i någon annan applikation som använder registret. T ex om posten används i Visma Administration.
1006	"Misslyckades att skapa en ny rad i registret"	Fel vid skapande av ny post i registret.
1007	"Misslyckades med att spara ändrad data."	Fel vid uppdateringen av en post.
1008	"Ej tillåtet att skapa fler än en instans av formuläret."	Endast tillåtet att öppna en koppling till registret.
1009	"Transaktionen redan startad."	Fel har uppstått i transaktionshanteringen.
1010	"Transaktionsfel."	Fel har uppstått i transaktionshanteringen.
1011	"Misslyckades med att låsa posten, posten redan låst."	Postlåsningsfel, posten är aktiv i någon annan applikation som använder registret. T ex om posten används i Visma Administration.
1012	"Ingen post finns."	Den post som angivits existerar inte.
1013	"Redigering inte tillåten i tabell"	Ej tillåtet att redigera i aktuellt register.
1014	"Nyckeln används redan"	Nyckelvärde måste vara unikt.
1015	"Post ej funnen"	Kan ej finna någon post som motsvarar detta sökvärde.
1016	"Blankt nyckelfält ej tillåtet"	Ett nyckelfält måste innehålla ett värde.
1017	"Registret är ej raderbart"	Kan ej radera aktuellt register.

1018	"Gick ej att skriva till transaktionsfil"	Fel har uppstått i transaktionshanteringen.
1019	"Misslyckades med att skriva ändrad data till databasfil"	Fel vid skrivningen av data till databasen.
1020	"Kan inte öppna. Företaget är öppnat exklusivt på en annan dator."	Företag kan öppnas exklusivt ifall man har endast en nätlicens, eller om företaget är markerat att det alltid ska öppnas exklusivt.
1021	"Kan inte öppna. Hela företaget är låst på en annan dator."	Någon har låst företaget för tex säkerhetskopiering.
1022	"Öppning av företaget misslyckades pga av låsningsproblem."	Övriga låsningsproblem.

Databas (IProgramPart: DB (=1))

101	Ett allvarligt fel har uppstått.
102	Inget felmeddelande angivet.
103	Det gick inte att initiera ODBC.
104	Det gick inte att hitta en lämplig ODBC-drivrutin.
105	Systemet stödjer inte rätt ODBC-version.
106	Det gick inte att ansluta till databasen.
107	Programmet har kopplats bort från databasen.
108	Databasen har inte stöd för operationen.
109	Tidsgränsen för databasanslutning överskreds.
110	Det gick inte att ladda ODBC-drivrutinen.
111	Transaktionen har redan rullats tillbaka.
112	Ingen transaktion är aktiv.
113	Markörtypen är inte giltig för operationen.
114	Ingen ytterligare data.
115	Det finns redan en databas med samma namn.
116	Filén hittades inte.
117	Fel vid läsning av DBF-fil.
118	Det gick inte att sätta värde på fältet.
119	Ogiltigt numeriskt värde för aktuellt fält.
120	Primärnyckel kan ej innehålla inkluderade kolumner.
121	Det angivna fältet finns inte.
122	Det går inte att utföra angiven databasjoin.
123	Programmet har förlorat kontakten med databasen.
124	Åtgärden har redan aktiverats.
125	Tabellen har inga nyckelfält.
126	Fält får inte vara NULL.
127	Posten hann ändrats av en annan användare.
128	Databasuppdateringen skulle ha påverkat fler än en rad.
129	Databasraden har raderats.
130	Duplikat nyckelvärde.
131	Inget felmeddelande angivet.
132	Ogiltigt värde.
133	Inget felmeddelande angivet.
134	Inget felmeddelande angivet.
135	Inget felmeddelande angivet.
136	Inget felmeddelande angivet.
137	Inget felmeddelande angivet.
138	Inget felmeddelande angivet.
139	Markören kunde inte positioneras på samma post som tidigare.
140	SQL-uttrycket är ogiltigt
141	Det angivna filtret är ogiltigt.
142	Användaren saknar behörighet att logga in.
143	Inloggning nekades.
144	Datatypen i databasfältet överensstämmer inte med av programmet förväntat värde.
145	Programmet kräver Microsoft SQL Server 2008 R2 eller senare.
146	Det gick inte att skapa en temporär fil. Kontrollera behörighet till tempkatalogen.
147	Kopplad post finns inte.
148	Databasstrukturen är felaktig. Uppgradering kan behövas.
149	Databasen behöver uppgraderas.
150	Filén finns redan.
151	Databasen och tempdb måste ha ALLOW_SNAPSHOT_ISOLATION påslaget
152	SQL Server har upptäckt ett I/O-fel. Hårddisken kan vara på väg att sluta fungera.

153	Ett problem hittades vid konvertering av databasen. Du måste kontakta Visma Spcs supportavdelning för att åtgärda detta.
154	Exakt träff hittades inte
155	Åtkomst nekad
156	Databasen är fylld. Kontrollera att det finns tillräckligt med diskutrymme.
157	Storleksgränsen på 10 GB per databas i SQL Server Express uppnåddes. Databasen är full.
158	Databasen har redan öppnats exklusivt och kan inte öppnas en gång till.
159	För lite systemminne tillgängligt för SQL Server.
160	Lagrad rutin finns inte.
161	En annan användare har gjort ändringar på data som förhindrar åtgärden från att slutföras.

AdkNet4Wrapper

Inledning

Vid installation av Visma Integration för Visma Administration 500/1000/2000 respektive Visma Integration för Visma Förening installeras AdkNet4Wrapper.dll i programkatalogen.

Information

AdkNet4Wrapper.dll använder adk.dll. För att AdkNet4Wrapper.dll ska fungera krävs en korrekt installation av Visma Administration/Visma Förening på samma dator. Med korrekt installation avses antingen serverinstallation, arbetsplatsinstallation eller en integrationsklientsinstallation.

Observera att AdkNet4Wrapper.dll är unik för Visma Administration respektive Visma Förening, dvs den som kommer med respektive program laddar bara adk.dll för sitt eget program.

AdkNet4Wrapper i sig är som huvudregel en transparent dll-fil som kapslar in Adk.dll och innehåller därför samma funktioner som finns i Adk.dll, det finns normalt inte heller någon skillnad på själva funktionaliteten. Detta innebär att informationen i Referensmanualen även gäller vid användandet av denna dll. Denna tilläggsmanual är enbart tänkt att visa hur man går till väga för att koppla samman en .NET-applikation med Visma Administration respektive Visma Förening.

Från version 2019.1 kommer vidareutveckling av stöd för .NET i API:et endast att göras i AdkNet4Wrapper.dll. From version 2019.1 finns dock vissa skillnader mot specifikationen av adk.dll, för att underlätta användningen utveckling i .NET. Se nedan för beskrivning av skillnaderna.

From version 2021.2 innehåller AdkNet4Wrapper.dll två namespaces, AdkNet4Wrapper och AdkNetWrapper.

Namespace samt ny funktion i AdkNetWrapper från version 2021.2

Namespace AdkNet4Wrapper innehåller Både nya versioner av funktioner och de gamla markerade som deprecated. Det är det namespace som funnits hela tiden i AdkNet4Wrapper.dll och finns därför kvar av bakåtkompatibilitetsskäl.

Namespace AdkNetWrapper innehåller bara de funktioner som inte är deprecated. Detta namespace är tänkt att användas i framtida versioner av vår NetWrapper när dessa anpassas till nya versioner av .NET. Detta namespace ska också säkerställas att överlagring av funktionsnamn inte krockar, framförallt i VB.Net.

I version 2021.2 har en ny funktion, `AdkSetTimestamp()`, lagts till för att hantera tidsstämpel med inparameter av typen `DateTime`. `AdkSetDate()` med samma inparametertyp kan bara hantera datumfält. Nya funktionen gör det möjligt att söka när man valt sökordning `eTimeStamp`. Funktionen finns tillgänglig i båda namespace.

Nyheter i AdkNet4Wrapper.dll i version 2019.1

Generellt har nya funktioner lagts till med stöd för .NET's datatyper. I normalfallet har det gjorts genom överlagring, men i vissa fall med helt nya funktioner. Gamla funktioner har markerats som obsoleta. Parametrar avsedda för utdata har deklarerats med `out`.

Följande funktioner har överlagrats med samma namn och dess parametrar har anpassats för .NET från version 2019.1

```
ADKERROR AdkAddEx(int pDb, bool bRow)
```

```

ADKERROR AdkFind2(int pDb, bool bRow)

ADKERROR AdkFindEx(int pDb, bool bRow)

ADKERROR AdkFirstEx(int pDb, bool bRow)

ADKERROR AdkGetBookingYearId(DateTime dValue, out string strId)

ADKERROR AdkGetBool(int pDb, int iFieldId, out bool pbValue)

ADKERROR AdkGetCompanyName(string pszFtgPath, bool bStrict, string
strSystemPath, out string chBuf, out int piVariant, out int lMajor, out int
lMinor)

ADKERROR AdkGetCompanyNameEx(string pszFtgPath, bool bStrict, string
strSystemPath, out string chBuf, out int piVariant, out int lMajor, out int
lMinor)

ADKERROR AdkGetCompanySetting(string pszSetting, out bool bValue)

ADKERROR AdkGetCustomerPrice(string strCustomer, string strArticle, double
dQuantity, bool bInvRab, bool bRowRab, bool bInclVAT,
ADK_CODE_OF_CUSTOMERPRICE iSelCur, out double dVal)

ADKERROR AdkGetCustomerPrice2(string strCustomer, string strArticle, double
dQuantity, bool bInvRab, bool bRowRab, bool bInclVAT,
ADK_CODE_OF_CUSTOMERPRICE iSelCur, out double dVal, out string ppsCurrency)

ADKERROR AdkGetDate(int pDb, int iFieldId, out DateTime? pdaValue)

int AdkGetDecimals(int pDb, int iFieldId)

string AdkGetErrorText(ADKERROR Error_Renamed, ADK_ERROR_TEXT_TYPE
nErrorTextType)

ADKERROR AdkGetFieldName(int IDataBaseId, int iFieldId, out string
ppsValue)

ADKERROR AdkGetSie(ADK_SIE_TYPE type, string strAar, string strFilename,
string strPath)

ADKERROR AdkGetStr(int pDb, int iFieldId, out string ppsValue)

ADKERROR AdkGetVariant(out int piVariant, out bool pbApi, out string
strBuf, string strSystemPath)

ADKERROR AdkGetVatpercentOnArticle(string strArticle, out double dVal)

ADKERROR AdkIsFilterable(int pDb, int iFieldId, out bool pbIsFilterable)

ADKERROR AdkIsLoginActivated(string pszSystemPath, string pszFtgPath, out
bool pbValue)

ADKERROR AdkLastEx(int pDb, bool bRow)

ADKERROR AdkNextEx(int pDb, bool bRow)

ADKERROR AdkOpen2(string pszSystemPath, string pszFtgPath, string
pszUserName, string pszPassword)

```

```

ADKERROR AdkOpenEx2(string pszSystemPath, string pszFtgPath, string
pszUserName, string pszPassword, int lLogOn);
AdkPreviousEx(int pDb, bool bRow)

ADKERROR AdkSetBool(int pDb, int iFieldId, bool bValue)

ADKERROR AdkSetDate(int pDb, int iFieldId, DateTime? lValue)

ADKERROR AdkSetFilter(int pDb, int iFieldId, string pExpression, int
iVersion)

ADKERROR AdkSetStr(int pDb, int iFieldId, string pValue)

ADKERROR AdkUpdateEx(int pDb, bool bRow)

```

Funktioner med avvikande returvärden mot hur de exponerats i Adk.dll fr o m version 2019.1

AdkGetDecimals

Motsvarande funktion i adk.dll returnerar felkod, den överlagrade returnerar enbart antal decimaler. Vid anrop med ickenumeriskt fält returneras 0, istället för felkod som gamla versionen av denna funktion gjorde. Likaså om pDb pekar mot ogiltig datastruktur.

AdkGetErrorText

Motsvarande funktion i adk.dll returnerar void, den överlagrade returnerar nu en sträng istf att resultatet skrevs till inparameter.

Följande funktioner finns from version 2019.2

Dessa funktioner är nya i adk.dll och i AdkNet4Wrapper.dll.

AdkDeleteRecord (ersätter AdkDelete)

```
ADKERROR AdkDeleteRecord(int pDb);
```

Följande funktioner är nya och ersätter gamla funktioner i AdkNet4Wrapper

Dessa funktioner finns inte i adk.dll.

AdkGetLastFieldId (ersätter AdkGetLastFldId)

```
ADKERROR AdkGetLastFieldId(int pDb, out int iFieldId);
```

AdkGetNrowsFieldId (ersätter AdkGetNrowsFldId)

```
ADKERROR AdkGetNrowsFieldId(int pDb, out int iFieldId);
```

AdkGetNumeric (ersätter AdkGetDouble)

```
ADKERROR AdkGetNumeric(int pDb, int iFieldId, out double pdValue);
ADKERROR AdkGetNumeric(int pDb, int iFieldId, out decimal pdValue);
```

AdkGetProgramVersion (ersätter AdkGetVersion)

```
void AdkGetProgramVersion(out int lMajor, out int lMinor);
```

AdkGetRowData (ersätter AdkGetData)

```
ADKERROR AdkGetRowData(int pParentDb, int iIndex, out int pDbRow);
```

AdkGetRowDataId (ersätter AdkGetRowDbId)

```
ADKERROR AdkGetRowDataId(int pDb, out int iRowDbId);
```

AdkGetRowsFieldId (ersätter AdkGetRowsFldId)

```
ADKERROR AdkGetRowsFieldId(int pDb, out int iFieldId);
```

AdkGetSupportedDatabaseVersion (ersätter AdkGetProgramDatabaseVersion)

```
void AdkGetSupportedDatabaseVersion(out int lMajor, out int lMinor);
```

AdkGetTimestampFieldId (ersätter AdkGetTimestampField)

```
ADKERROR AdkGetTimestampFieldId(int pDb, out int iFieldId);
```

Funktioner med samma definition som tidigare versioner av AdkNet4Wrapper fr o m version 2019.1

```
ADKERROR AdkAdd(int pDb)
```

```
ADKERROR AdkClearFilters(int pDb)
```

```
int AdkClose()
```

```
int AdkCreateData(int IDataBaseId)
```

```
int AdkCreateDataRow(int IDataBaseId, int iNumber)
```

```
ADKERROR AdkDeleteRow(int pDb, int iIx)
```

```
ADKERROR AdkDeleteStruct(int pDb)
```

```
ADKERROR AdkFind(int pDb)
```

```
ADKERROR AdkFirst(int pDb)
```

```
int AdkGetDataRow(int pDb, int iIndex)
```

```
ADKERROR AdkGetLength(int pDb, int iFieldId, ref int iLength)
```

```
ADKERROR AdkGetType(int pDb, int iFieldId, ref ADK_FIELD_TYPE pType)
```

```
bool AdkIsReadWrite(int pDb, int iFieldId)
```

```
ADKERROR AdkLast(int pDb)
```

```
ADKERROR AdkNext(int pDb)
```

```
ADKERROR AdkPrevious(int pDb)
```

```
ADKERROR AdkResetStruct(int pDb)
```

```
ADKERROR AdkSetData(int pDb, int iFieldId, int pDataRow)
```

```
ADKERROR AdkSetDouble(int pDb, int iFieldId, double dValue)
```

```
ADKERROR AdkSetSortOrder(int pDb, ADK_SORT_ORDER nSortOrder)
```

```
void AdkSetWarningDelegate(AdkWarningDelegate pFunction)
```

```
ADKERROR AdkUpdate(int pDb)
```


Sökväg till gemensamma filer respektive företaget i AdkNet4Wrapper från version 2019.1

Vid användning av AdkNet4Wrapper.dll är det inte nödvändigt att ange sökväg till katalogen för gemensamma filer vid anrop till funktioner som tar denna som inparameter. Om sökvägen lämnas blank letar AdkNet4Wrapper.dll upp rätt sökväg i windows register.

Vid användning av AdkNet4Wrapper är det inte nödvändigt att ange hela sökvägen till företagskatalogen, utan en relativ sökväg till standardsökvägen för företag kan anges. Ex Ovnbol2000 gör att Apiet hittar Installerat övningsbolag för Visma Administration 2000.

Läs avsnittet Registerinformation i början av referensmanualen vad installationen lägger upp för information.

För praktisk demonstration, provkör och öppna företag i demoprojektet (DemoApiCSharp) som medföljer utvecklarpaketet.

Följande funktioner omfattas av denna funktionalitet:

```
AdkGetCompanyName  
AdkGetCompanyNameEx  
AdkGetVariant  
AdkIsLoginActivated  
AdkOpen2  
AdkOpen2Ex
```

Användning av AdkNet4Wrapper

Följande exempel är huvudsakligen gjorda i C#, men dess motsvarigheter i övriga språk med stöd för .NET fungerar på likartat sätt.

Denna tilläggsmanual är som ovan beskrivits ett komplement till den vanliga Referensmanualen. Den ordinarie Referensmanualen bör användas för att få ytterligare information om de nedan beskrivna funktionerna samt funktioner som ej omnämns i detta dokument.

Demoprojektet för C# för .NET 4 (DemoApiCSharp) i utvecklarpaketet har uppdaterats att använda nyheterna i AdkNet4Wrapper, så ytterligare kodexempel finns där.

Komma igång

För att använda sig av AdkNet4Wrapper måste man ha en referens till dll-filen. Detta görs genom att i projektet referera till AdkNet4Wrapper.dll.

Felhantering

En viktig del i utvecklandet är att ta hand om felkoder som genereras av API:et, vilket görs på följande sätt.

```
using AdkNet4Wrapper;  
  
Api.ADKERROR error;  
...  
if (error.lRc != Api.ADKE_OK)  
{  
    String errortext = Api.AdkGetErrorText(error,  
        Api.ADK_ERROR_TEXT_TYPE.elRc);  
    MessageBox.Show(errortext);  
}
```

```
}
```

Som synes i exempel ovan måste man ha en variabel av typen ADKERROR i vilken man lägger resultatet från exekverade funktioner. I detta exempel visas ett felmeddelande om felkoden skiljer sig från OK och en meddelanderuta visas som innehåller felmeddelandet.

Öppna företag

Följande kod visar hur man öppnar ett företag via AdkNet4Wrapper och visar ett felmeddelande om försöket att öppna företaget skulle misslyckas.

```
using AdkNet4Wrapper;

Api.ADKERROR error;

String sys = @"C:\ProgramData\SPCS\SPCS Administration\Gemensamma filer";
String ftg = @"C:\ProgramData\SPCS\SPCS Administration\Företag\Ovnbol2000";
String user = "";
String pwd = "";

error = Api.AdkOpen2(sys, ftg, user, pwd);
if (error.lRc != Api.ADKE_OK)
{
    String errortext = Api.AdkGetErrorText(error,
        Api.ADK_ERROR_TEXT_TYPE.elRc);
    MessageBox.Show(errortext);
}
```

Detta exempel utgår från att Visma Administration används och att programmet är installerat på standardsökväg.

Stäng företag

Vid stängning av företag används AdkClose. Detta ska göras för att företaget ska stängas på ett korrekt sätt.

```
Api.AdkClose();
```

Skapa datastruktur

De flesta funktioner kräver att man skapat en datastruktur där resultatet kan placeras, motsvarande en rad i någon av tabellerna. Detta görs med hjälp av funktionen AdkCreateData.

Detta görs genom följande:

```
int pData = Api.AdkCreateData(Api.ADK_DB_CUSTOMER);
```

I det här exemplet skapas en datastruktur till kundtabellen.

Hämta/läsa värden från datastruktur

För att läsa ut ett värde används någon av de olika AdkGet-funktionerna.

```
String info;
error = Api.AdkGetStr(pData, Api.ADK_CUSTOMER_NAME, out info);
```

I det här exemplet håller variabeln pData själva datastrukturen och man försöker läsa kundnamnet från kundtabellen. Först måste dock någon post läst in, exempelvis via AdkFind eller AdkFirst. Ett mer komplett exempel på att hämta ut ett värde och visa värdet i en dialogruta ses nedan.

```
using AdkNet4Wrapper;

Api.ADKERROR error;
int pData;

pData = Api.AdkCreateData(Api.ADK_DB_CUSTOMER);

error = Api.AdkFirst(pData);
if (error.lRc != Api.ADKE_OK)
{
    String errortext = Api.AdkGetErrorText(error,
        Api.ADK_ERROR_TEXT_TYPE.elRc);
    MessageBox.Show(errortext);
}

String info;
error = Api.AdkGetStr(pData, Api.ADK_CUSTOMER_NAME, out info);
if (error.lRc != Api.ADKE_OK)
{
    String errortext = Api.AdkGetErrorText(error,
        Api.ADK_ERROR_TEXT_TYPE.elRc);
    MessageBox.Show(errortext);
}

MessageBox.Show(info);
```

Sätta/skriva värden till datastruktur

För att sätta ett värde till en datastruktur ska man använda sig av AdkSet-funktionerna.

```
String info = "Testkund";
error = Api.AdkSetStr(pData, Api.ADK_CUSTOMER_NAME, info);
```

I det här exemplet pekar pData mot kundtabellen och man försöker skriva till kundnamnsfältet. Ett mer komplett exempel på att sätta kundnamnet syns nedan.

```
using AdkNet4Wrapper;

Api.ADKERROR error;
int pData;

pData = Api.AdkCreateData(Api.ADK_DB_CUSTOMER);

String info = "Testkund";
error = Api.AdkSetStr(pData, Api.ADK_CUSTOMER_NAME, info);
if (error.lRc != Api.ADKE_OK)
{
    String errortext = Api.AdkGetErrorText(ref error,
        Api.ADK_ERROR_TEXT_TYPE.elRc);
    MessageBox.Show(errortext);
}
```

Lägg till data (post)

För att skapa en ny post, exempelvis en ny kund, använder man sig av AdkAdd och skickar in datastrukturen som ska bilda den nya posten. Enbart de värden som blivit uppdaterade i strukturen används vid skapandet av posten. Exempel på att skapa en kund kan ses nedan.

```
using AdkNet4Wrapper;

Api.ADKERROR error;
int pData;

pData = Api.AdkCreateData(Api.ADK_DB_CUSTOMER);

String info = "Testkund";
error = Api.AdkSetStr(pData, Api.ADK_CUSTOMER_NAME, info);
if (error.lRc != Api.ADKE_OK)
{
    String errortext = Api.AdkGetErrorText(error,
        Api.ADK_ERROR_TEXT_TYPE.elRc);
    MessageBox.Show(errortext);
}

error = AdkNetWrapper.Api.AdkAdd(pData);
if (error.lRc != Api.ADKE_OK)
{
    String errortext = Api.AdkGetErrorText(error,
        Api.ADK_ERROR_TEXT_TYPE.elRc);
    MessageBox.Show(errortext);
}
```

Datatyper

Det finns ett antal olika datatyper (enumerationer och datastrukturer) definierade för API:et och även dessa är åtkomliga via AdkNet4Wrapper.dll.

Exempel på praktisk användning kan ses nedan, där man väljer vilket felmeddelande man vill komma åt, i detta fall själva feltexten.

```
int errtype = Api.ADK_ERROR_TEXT_TYPE.elRc;
String errortext = Api.AdkGetErrorText(error, errtype);
MessageBox.Show(errortext);
```

Felkoder

De felkoder som finns i API:et finns definierade även i AdkNetWrapper.

Exempel för att använda felkoden OK är

```
Adk4NetWrapper.Api.ADKE_OK
```

Ett mer konkret exempel på hur detta kan utnyttjas ses nedan där man går igenom hela kundregistret.

```
using AdkNet4Wrapper;

Api.ADKERROR error;
int pData;
```

```

pData = Api.AdkCreateData(Api.ADK_DB_CUSTOMER);

error = AdkNetWrapper.Api.AdkFirst(pData);
if (error.lRc != Api.ADKE_OK)
{
    String errortext = Api.AdkGetErrorText(error,
        Api.ADK_ERROR_TEXT_TYPE.elRc);
    MessageBox.Show(errortext);
}

while (error.lRc != Api.ADKE_EOF)
{
    error = Api.AdkNext(pData);
}

```

AdkSetWarningDelegate

AdkSetWarningFunction har gjorts tillgänglig med hjälp av AdkSetWarningDelegate. AdkSetWarningFunction används för att peka ut en funktion som ska hantera varningar genererade i programmet. Se avsnittet AdkSetWarningFunction för detaljer.

Det är viktigt att tänka på garbage collection när man använder AdkSetWarningDelegate.

Följande konstruktion bör användas:

```

myGlobalLoggWarningDelegate = new Api.AdkWarningDelegate(MyLoggFunction);
Api.AdkSetWarningDelegate(myGlobalLoggWarningDelegate);

```

Gör man ett anrop till AdkSetWarningDelegate där man skapar objektet direkt i anropet riskerar man att garbage collection slår till och får problem när vårt API genererar en varning.

AdkNetWrapper

Inledning

OBS! Från version 2019.1 kommer ingen vidareutveckling att göras av AdkNetWrapper.dll. Versionen som skickas med i installationen är densamma som levererades med version 2019.0 av programmet.

Från version 4.4 av Visma Integration bifogas en ny dll-fil vilken är tänkt att användas ihop med utveckling i .NET (version 2.0 – 3.5 SP1). Dll-filen kan inte användas separat utan är ett tillägg till Visma Integration.

Vid installation av Visma Integration för Visma Administration 500/1000/2000 respektive Visma Integration för Visma Förening installeras även denna dll för att säkerställa att rätt version finns installerad.

Information

Dll-filen heter AdkNetWrapper.dll och ska ligga i samma mapp som Adk.dll. Wrappern i sig är en transparent dll-fil och innehåller därför inga funktioner som inte finns i Adk.dll, det finns inte heller någon skillnad på själva funktionaliteten. Detta innebär att informationen i Referensmanualen även gäller vid användandet av denna dll. Denna tilläggsmanual är enbart tänkt att visa hur man går till väga för att koppla samman en .NET-applikation med Visma Administration respektive Visma Förening.

Användning av AdkNetWrapper

Följande exempel är huvudsakligen gjorda i C#, men dess motsvarigheter i övriga språk med stöd för .NET fungerar på likartat sätt.

Denna tilläggsmanual är som ovan beskrivits ett komplement till den vanliga Referensmanualen. Den ordinarie Referensmanualen bör användas för att få ytterligare information om de nedan beskrivna funktionerna samt funktioner som ej omnämns i detta dokument.

Komma igång

För att använda sig av AdkNetWrapper måste man ha en referens till dll-filen. Detta görs genom att i projektet referera till AdkNetWrapper.dll.

Felhantering

En viktig del i utvecklandet är att ta hand om felkoder som genereras av API:et, vilket görs på följande sätt.

```
AdkNetWrapper.Api.ADKERROR error;
...
if (error.lRc != AdkNetWrapper.Api.ADKE_OK)
{
    String errortext = new String(' ', 200);
    int errtype = (int)AdkNetWrapper.Api.ADK_ERROR_TEXT_TYPE.elRc;
    AdkNetWrapper.Api.AdkGetErrorText(ref error, errtype,
        ref errortext, 200);
    MessageBox.Show(errortext);
}
```

```
}
```

Som synes i exempel ovan måste man ha en variabel av typen ADKERROR i vilken man lägger resultatet från exekverade funktioner. I detta exempel visas ett felmeddelande om felkoden skiljer sig från OK och en meddelanderuta visas som innehåller felmeddelandet.

Öppna företag

Följande kod visar hur man öppnar ett företag via AdkNetWrapper och visar ett felmeddelande om försöket att öppna företaget skulle misslyckas.

```
AdkNetWrapper.Api.ADKERROR error;

String sys = @"C:\ProgramData\SPCS\SPCS Administration\Gemensamma filer";
String ftg = @"C:\ProgramData\SPCS\SPCS Administration\Företag\Ovnbol2000";

error = AdkNetWrapper.Api.AdkOpen(ref sys, ref ftg);
if (error.lRc != AdkNetWrapper.Api.ADKE_OK)
{
    String errortext = new String(' ', 200);
    int errtype = (int)AdkNetWrapper.Api.ADK_ERROR_TEXT_TYPE.elRc;
    AdkNetWrapper.Api.AdkGetErrorText(ref error, errtype,
        ref errortext, 200);
    MessageBox.Show(errortext);
}
```

Detta exempel utgår från att Visma Administration används och att programmet är installerat på standardsökväg.

Stäng företag

Vid stängning av företag används AdkClose. Detta ska göras för att företaget ska stängas på ett korrekt sätt.

```
AdkNetWrapper.Api.AdkClose();
```

Skapa datastruktur

De flesta funktioner kräver att man skapat en datastruktur där resultatet kan placeras, motsvarande en rad i någon av tabellerna. Detta görs med hjälp av funktionen AdkCreateData.

Detta görs genom följande:

```
int pData;
pData = AdkNetWrapper.Api.AdkCreateData(AdkNetWrapper.Api.ADK_DB_CUSTOMER);
```

I det här exemplet skapas en datastruktur till kundtabellen.

Hämta/läsa värden från datastruktur

För att läsa ut ett värde används någon av de olika AdkGet-funktionerna.

```
String info = new String(' ', 50);
error = AdkNetWrapper.Api.AdkGetStr(pData,
    AdkNetWrapper.Api.ADK_CUSTOMER_NAME, ref info, 50);
```

I det här exemplet håller variabeln pData själva datastrukturen och man försöker läsa kundnamnet från kundtabellen. Först måste dock någon post läst in, exempelvis via AdkFind eller AdkFirst. Ett mer komplett exempel på att hämta ut ett värde och visa värdet i en dialogruta ses nedan.

```
AdkNetWrapper.Api.ADKERROR error;
int pData;

pData = AdkNetWrapper.Api.AdkCreateData(AdkNetWrapper.Api.ADK_DB_CUSTOMER);

error = AdkNetWrapper.Api.AdkFirst(pData);
if (error.lRc != AdkNetWrapper.Api.ADKE_OK)
{
    String errortext = new String(' ', 200);
    int errtype = (int)AdkNetWrapper.Api.ADK_ERROR_TEXT_TYPE.elRc;
    AdkNetWrapper.Api.AdkGetErrorText(ref error, errtype, ref errortext,
200);
    MessageBox.Show(errortext);
}

String info = new String(' ', 50);
error = AdkNetWrapper.Api.AdkGetStr(pData,
AdkNetWrapper.Api.ADK_CUSTOMER_NAME, ref info, 50);
if (error.lRc != AdkNetWrapper.Api.ADKE_OK)
{
    String errortext = new String(' ', 200);
    int errtype = (int)AdkNetWrapper.Api.ADK_ERROR_TEXT_TYPE.elRc;
    AdkNetWrapper.Api.AdkGetErrorText(ref error, errtype, ref errortext,
200);
    MessageBox.Show(errortext);
}

MessageBox.Show(info);
```

Sätta/skriva värden till datastruktur

För att sätta ett värde till en datastruktur ska man använda sig av AdkSet-funktionerna.

```
String info = "Testkund";
error = AdkNetWrapper.Api.AdkSetStr(pData,
AdkNetWrapper.Api.ADK_CUSTOMER_NAME, ref info);
```

I det här exemplet pekar pData mot kundtabellen och man försöker skriva till kundnamnsfältet. Ett mer komplett exempel på att sätta kundnamnet syns nedan.

```
AdkNetWrapper.Api.ADKERROR error;
int pData;

pData = AdkNetWrapper.Api.AdkCreateData(AdkNetWrapper.Api.ADK_DB_CUSTOMER);

String info = "Testkund";
error = AdkNetWrapper.Api.AdkSetStr(pData,
AdkNetWrapper.Api.ADK_CUSTOMER_NAME, ref info);
if (error.lRc != AdkNetWrapper.Api.ADKE_OK)
{
    String errortext = new String(' ', 200);
    int errtype = (int)AdkNetWrapper.Api.ADK_ERROR_TEXT_TYPE.elRc;
    AdkNetWrapper.Api.AdkGetErrorText(ref error, errtype, ref errortext,
200);
    MessageBox.Show(errortext);
}
```



```
}
```

Lägg till data (post)

För att skapa en ny post, exempelvis en ny kund, använder man sig av AdkAdd och skickar in datastrukturen som ska bilda den nya posten. Enbart de värden som blivit uppdaterade i strukturen används vid skapandet av posten. Exempel på att skapa en kund kan ses nedan.

```
AdkNetWrapper.Api.ADKERROR error;
int pData;

pData = AdkNetWrapper.Api.AdkCreateData(AdkNetWrapper.Api.ADK_DB_CUSTOMER);

String info = "Testkund";
error = AdkNetWrapper.Api.AdkSetStr(pData,
AdkNetWrapper.Api.ADK_CUSTOMER_NAME, ref info);
if (error.lRc != AdkNetWrapper.Api.ADKE_OK)
{
    String errortext = new String(' ', 200);
    int errtype = (int)AdkNetWrapper.Api.ADK_ERROR_TEXT_TYPE.elRc;
    AdkNetWrapper.Api.AdkGetErrorText(ref error, errtype, ref errortext,
200);
    MessageBox.Show(errortext);
}

error = AdkNetWrapper.Api.AdkAdd(pData);
if (error.lRc != AdkNetWrapper.Api.ADKE_OK)
{
    String errortext = new String(' ', 200);
    int errtype = (int)AdkNetWrapper.Api.ADK_ERROR_TEXT_TYPE.elRc;
    AdkNetWrapper.Api.AdkGetErrorText(ref error, errtype, ref errortext,
200);
    MessageBox.Show(errortext);
}
```

Datatyper

Det finns ett antal olika datatyper (enumerationer och datastrukturer) definierade för API:et och även dessa är åtkomliga via AdkNetWrapper.dll.

Exempel på praktisk användning kan ses nedan, där man väljer vilket felmeddelande man vill komma åt, i detta fall själva feltexten.

```
String errortext = new String(' ', 200);
int errtype = (int)AdkNetWrapper.Api.ADK_ERROR_TEXT_TYPE.elRc;
AdkNetWrapper.Api.AdkGetErrorText(ref error, errtype, ref errortext, 200);
MessageBox.Show(errortext);
```

Felkoder

De felkoder som finns i API:et finns definierade även i AdkNetWrapper.

Exempel för att använda felkoden OK är

```
AdkNetWrapper.Api.ADKE_OK
```

Ett mer konkret exempel på hur detta kan utnyttjas ses nedan där man går igenom hela kundregistret.

```
AdkNetWrapper.Api.ADKERROR error;
int pData;

pData = AdkNetWrapper.Api.AdkCreateData(AdkNetWrapper.Api.ADK_DB_CUSTOMER);

error = AdkNetWrapper.Api.AdkFirst(pData);
if (error.lRc != AdkNetWrapper.Api.ADKE_OK)
{
    String errortext = new String(' ', 200);
    int errtype = (int)AdkNetWrapper.Api.ADK_ERROR_TEXT_TYPE.elRc;
    AdkNetWrapper.Api.AdkGetErrorText(ref error, errtype, ref errortext,
200);
    MessageBox.Show(errortext);
}

while (error.lRc != AdkNetWrapper.Api.ADKE_EOF)
{
    error = AdkNetWrapper.Api.AdkNext(pData);
}
```

AdkSetWarningDelegate

AdkSetWarningFunction har gjorts tillgänglig med hjälp av AdkSetWarningDelegate.

AdkSetWarningFunction används för att peka ut en funktion som ska hantera varningar genererade i programmet. Se avsnittet AdkSetWarningFunction för detaljer.

Det är viktigt att tänka på garbage collection när man använder AdkSetWarningDelegate.

Följande konstruktion bör användas:

```
myGlobalLoggWarningDelegate = new Api.AdkWarningDelegate(MyLoggFunction);
Api.AdkSetWarningDelegate(myGlobalLoggWarningDelegate);
```

Gör man ett anrop till AdkSetWarningDelegate där man skapar objektet direkt i anropet riskerar man att garbage collection slår till och får problem när vårt API genererar en varning.