

- [BI Populator Command Line Arguments](#)
- [dmUser SysAdmin Setting](#)
- [Spread Logic may Cause Large Spread Amount](#)
- [BI Populator Object Id Filter Mapping](#)
- [shBIInterface Function Codes](#)

## BI Populator Command Line Arguments

---

The following command line arguments may be used to run the BI Populator from the command line prompt or to schedule the BI Populator to run using Schedule Admin.

In the table, parameters are enclosed to indicate information that needs to be supplied by the user. When entering the actual parameter (e.g., the user name), do not use the <> symbols. For example, to enter the user name and password for a particular user and schedule an automatic run, you would type the following.

```
-U username -P password -auto -date 01/01/2006
```

Argument	Description
-U <username>	Identifies the user for automatic login. Enter the AdBase User name to login.
-P <password>	Provides the password for automatic login. Enter the AdBase password to login.
-auto	Runs the BI Populator in automatic mode. This means that the BI Populator executes automatically without a user having to click the Populate button. Once processing is complete, the BI Populator will automatically exit. This argument is primarily for use in Schedule Admin.
-continuous	Runs the BI Populator in automatic mode. This means that the BI Populator executes automatically without a user having to click the Populate button. Unlike the -auto argument, once the process is complete, the BI Populator continues to
-date <start date>	Specifies the start date in the absolute date format (mm/dd/yyyy).
-D <# days in future>	Specifies the start date as the number of days from the current date.
-duration <# days>	Specifies the end date as the number of days after the start date.
-crdate	Specifies selection by the create date, rather than the last edit date. Using this argument is essentially the same as checking the Use create date box in the BI Populator window. When using this argument, do not specify a date. NOTE: If you do not use the crdate or ledate argument in the command line, BI Populator will run using whichever setting was selected the last time the utility was exited.
-ledate	Specifies selection by the last edit date, rather than the create date. Using this argument is essentially the same as checking the Use last edit date box in the BI Populator window. When using this argument, do not specify a date.
-customer	Specifies customer population will be performed.
-customerfile <filepath>	Specifies a file containing a list of customer IDs to be populated. Note that if you use this argument, you do not have to use the -customer argument. The presence of the -customerfile argument specifies customer population will be performed.
-contract	Specifies contract population will be performed.

Argument	Description
-contractfile <file path>	Specifies a file containing a list of contract IDs to be populated. Note that if you use this argument, you do not have to use the -contract argument. The presence of the -contractfile argument specifies that contract population will be performed.
-gl	Specifies GL population will be performed.
-invoicefile <file path>	Specifies a file containing a list of invoice IDs to be populated. Note that if you use this argument, you do not have to use the -gl argument. The presence of the -invoicefile argument specifies that invoice population will be performed.
-debitfile <file path>	Specifies a file containing a list of debit IDs to be populated. Note that if you use this argument, you do not have to use the -gl argument. The presence of the -debitfile argument specifies that debit population will be performed.
-paymentfile <file path>	Specifies a file containing a list of payment IDs to be populated. Note that if you use this argument, you do not have to use the -gl argument. The presence of the -paymentfile argument specifies that payment population will be performed.
-creditfile <file path>	Specifies a file containing a list of credit IDs to be populated. Note that if you use this argument, you do not have to use the -gl argument. The presence of the -creditfile argument specifies that credit population will be performed.
-adorder	Specifies ad order population will be performed.
-adorderfile <file path>	Specifies a file containing a list of ad order IDs to be populated. Note that if you use this argument, you do not have to use the -adorder argument. The presence of the -adorderfile argument specifies that credit population will be performed.
-leavesettings	If you run two copies of the BI Populator, they interfere with each other's settings. A copy of the BI Populator called with this argument will avoid updating the settings, leaving the original copy of the BI Populator alone.

# dmUser SysAdmin Setting

A setting in System Admin → Tools/System Level Information → Other Settings tab called “Update Existing dmUser Entry” is used to control when a user is moved to a new sales team, region, territory or company.

Financial Settings	Rating Settings	Other Settings	Web Settings	Customer Settings	Ad Orders	Ad Orders (Cont)	Preprints	Campaigns	B
--------------------	-----------------	----------------	--------------	-------------------	-----------	------------------	-----------	-----------	---

User Settings

☐ Use Password Aging: Interval Days   
☒ Digit Required in Password  
☐ Mixed Case Required in Password  
☒ Disable After Failed Login  
☒ Skip auto logoff if object on display  
  
Minimum Password Length   
Auto Logoff Minutes   
Default Max Locations

Days

☐ Windows Authentication Login  
☐ Network Login Is Login Name  
Default Domain   

Require Login

Custom Colors

R :	<input type="text" value="255"/>	R :	<input type="text" value="255"/>	R :	<input type="text" value="255"/>	R :	<input type="text" value="255"/>	R :	<input type="text" value="255"/>
G :	<input type="text" value="255"/>	G :	<input type="text" value="255"/>	G :	<input type="text" value="255"/>	G :	<input type="text" value="255"/>	G :	<input type="text" value="255"/>
B :	<input type="text" value="255"/>	B :	<input type="text" value="255"/>	B :	<input type="text" value="255"/>	B :	<input type="text" value="255"/>	B :	<input type="text" value="255"/>
	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>
R :	<input type="text" value="255"/>	R :	<input type="text" value="255"/>	R :	<input type="text" value="255"/>	R :	<input type="text" value="255"/>	R :	<input type="text" value="255"/>
G :	<input type="text" value="255"/>	G :	<input type="text" value="255"/>	G :	<input type="text" value="255"/>	G :	<input type="text" value="255"/>	G :	<input type="text" value="255"/>
B :	<input type="text" value="255"/>	B :	<input type="text" value="255"/>	B :	<input type="text" value="255"/>	B :	<input type="text" value="255"/>	B :	<input type="text" value="255"/>

Order Quote Info

Quote

Quote Order Status

LAN Settings

☐ Use Proxy Server for LAN Address

Contracts

☐ Allow New Duplicate Instance Names

Typographic

Max Leading in pts 

Auto

Customer Transactions

☒ Exposure ☐ Search  
☐ Invoice ☐ Balance History  
☐ Documents ☐ Balance Info

Measurement Units

Inches

Graphics Display

Blob Purge Date Days

Data  Html  Eps

Business Intelligence

☒ Enable BI Feeder Table Processing  
☐ Update Existing dmUser Entry  
☐ Spread Schedule Level Charges  
☐ Spread Tax Charges  
☐ No Spread Special Discounts  
☐ No Spread Special Price When As Run

Output Parameters

EPS

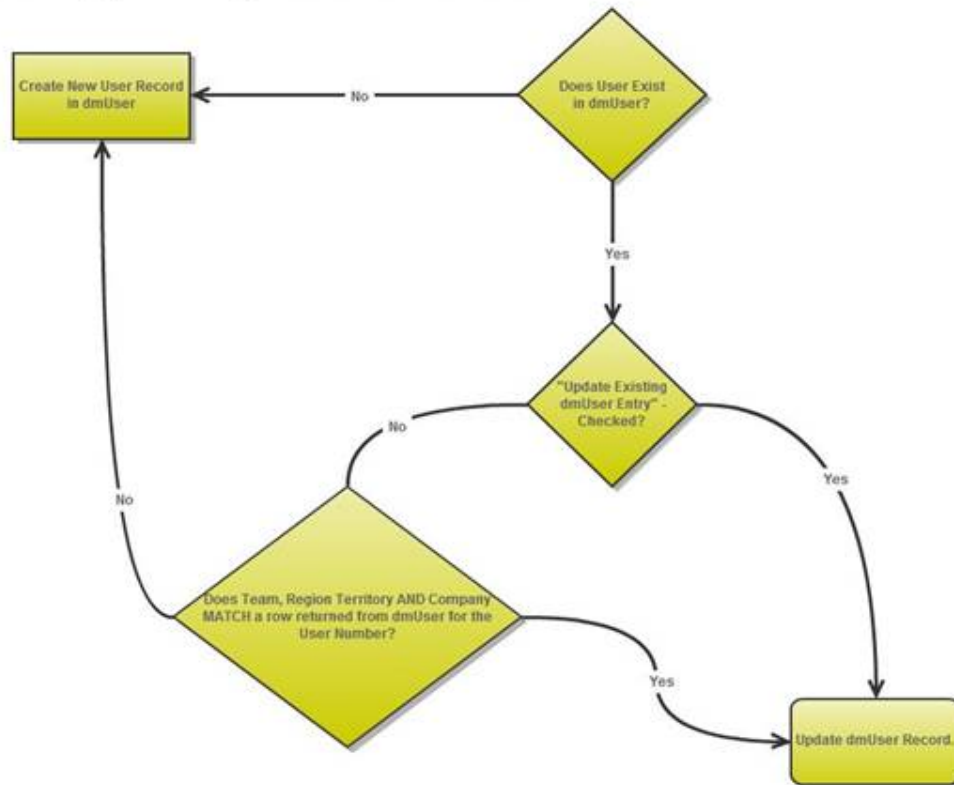
## Update Existing dmUser Entry - CHECKED

When this option is checked, any change made to the sales rep's Team, Region, Territory or Company within system admin is reflected by updating the existing row in the dmUser table.

## Update Existing dmUser Entry - UNCHECKED

If this option is not checked, than when a change is made to a sales rep's Team, Region, Territory or Company a new row is created in dmUser for that sales rep, but his old row still exists.

# Update Existing dmUser Entry Setting in SystemAdmin



## What Does This Mean

These options affect how the Sold By rep's transactions are able to be reported.

Every transaction (ad taken) has a Sold By sales rep attached to it. This is a field that is set in Adbooker.

When this option is Unchecked it will effectively let the Sold By rep's transactions "move" with him when he moves to a new sales team, territory or region.

For example, Rep JDoe is on Team Alpha in '17 and in Dec '17 he sold 10 ads in which he was the rep in the Sold By field in Ad Booker. If we were to run a report for Dec '17 for Team Alpha, Rep JDoe's sales would be included in the total.

## Update Existing dmUser Entry - CHECKED

If in Jan '18 Rep JDoe was moved to Team Beta and Option 1 was set in System Admin, the same report, Dec '17 for Team Alpha, Rep JDoe's sales would NO longer be included in the total.

## Update Existing dmUser Entry - UNCHECKED

However if this option is checked in System Admin, the above report would be the same both before and after Rep JDoe was moved to Team Beta.

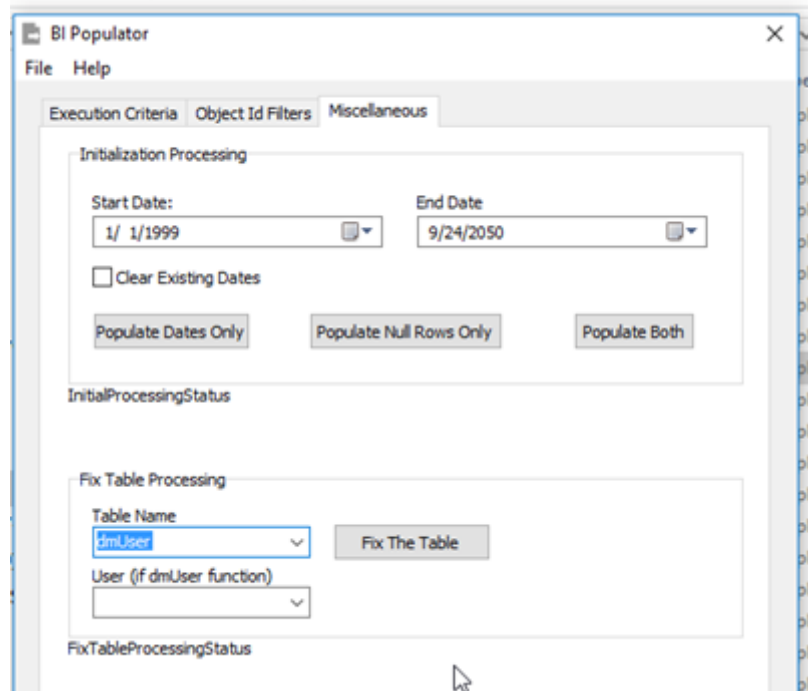
Be aware the Rep JDoe always is the Sold By rep on the ads. This just allows you to choose what Team, Region, Territory or Company those ads should show up in.

## BE CAREFUL Changing This Option

If you have been populating the BI Database with this option unchecked and then decide you want it checked. You will need to run a process in the BI Populator to make sure the data is consistent.

If you change this setting, from being un-checked to being checked, there is a function in BiPopulator to update the existing entries in BI. This would merge all entries for a sales rep to point to one record in BI.

Some sites start with the setting unchecked, but realize they want to change this and go to one record per sales rep. To update existing data to have one row vs. multiple you can use a function in BI Populator:



You click on 'Fix The Table' and it goes through all the [dmUser](#) records and merges them into one per sales rep. **Note:** Always do this in a **separate** BI Populator session here you are logged in with a user that is NOT the same as the Automated BI Populator.

---

# Spread Logic may Cause Large Spread Amount

First seen in 2009

First and foremost this is not a bug in the sense that there is an error in the code. As you will see it is caused by how the spread is calculated.

The spread algorithm is designed to spread an amount, like order level charges, credits, etc, across all other charges and distribute a weighted amount to each.

Here is how the algorithm figures the weights:

Assume we are spreading a \$50 Order Level charge across an ad with 3 revenue entries

<b>Amount To Spread</b>	<b>50</b>
-------------------------	-----------

<b>Date</b>	<b>Amount</b>	<b>Calc Weight Line Amt/Total Amt</b>	<b>Calced Weight</b>	<b>% Needed of Amt To Spread</b>	<b>Spread Amt</b>
4/1/2009	\$ 10.00	10/30	0.333333	33%	16.67
4/2/2009	\$ 15.00	15/30	0.500000	50%	25.00
4/3/2009	\$ 5.00	5/30	0.166667	17%	8.33
<b>Total</b>	<b>\$ 30.00</b>		<b>1.000000</b>	<b>100%</b>	<b>50</b>

1. Add all the revenue amounts together that we are going to spread the order level charge across. (Amount Column)
2. Divide each line item amount by the total amount to get a weighted percentage of how much of the spread amount to allocate to this line item.
3. Multiply the Spread Amount by the calculated percentage found in step 2 to get the spread amount allocation.

This works fine until you run into a situation with line items as follows:

<b>Amount To Spread</b>	<b>-400</b>
-------------------------	-------------

<b>Date</b>	<b>Amount</b>	<b>Calc Weight Line Amt/Total Amt</b>	<b>Calced Weight</b>	<b>% Needed of Amt To Spread</b>	<b>Spread Amt</b>
4/1/2009	\$ 0.50	.5/.1	5	500%	-2000
4/1/2009	\$ 0.50	.5/.1	5	500%	-2000
4/1/2009	\$ (90.00)	-90/0.1	-900	-90000%	360000
4/1/2009	\$ 89.10	89.10/.1	891	89100%	-356400
<b>Total</b>	<b>\$ 0.10</b>				<b>-400</b>

You can see that because the final total is a small number our percentages are very high, but in the end add to a total of 100% and the spread amount add to the correct number.

This is most commonly seen in Till forbid ads.

# BI Populator Object Id Filter Mapping

The BI Populator allows you to manually populate individual records from the Object Id Filters tab. Below is a mapping to the fields that are being referenced on this tab.

The screenshot shows the 'BI Populator' application window with the 'Object Id Filters' tab selected. The window contains several sections for different data types, each with checkboxes for 'Get [type] ids from file' and 'Use [type] id range', and a 'Browse' button. To the right of each section, a text box indicates the corresponding database field.

Section	Field
Customers	CUSTOMER.ACCOUNTID
Contracts	COCONTRACTINSTANCE.ID
Invoices	FNTRANSACTION.ID
Debits	FNTRANSACTION.ID
Payments	AOPAYMENTS.TRANSID
Credits	AOCUSTOMERCD.TRANSID
Ad Orders	AOADORDER.ID

## shBIInterface Function Codes

The shBIInterface table in the AdBase core database is a table used to tell the BI Populator to populate information that will not get picked up by the BI Populator's normal check.

Usually, it is utility applications like RepSetter that will populate this table when it is run by users.

```
select * from shbiInterface
```

One of the fields in the shBIInterface table is the `FunctionCode` field. This code represents what type of populating is going to occur. The other field of importance is the `ApplicationId` which tells us which application wrote the record in the shBIInterface table. Lastly, the `ObjectId`, is the associated ID to be populated. Whether the `ObjectId` represents an `AdOrderId` or `CustomerId` or something else, depends on the `FunctionCode`.



Below is a list of the function codes and their enumerations.

```
#ifndef ShBIInterfaceEnumTypes_h #define ShBIInterfaceEnumTypes_h 1

typedef enum { ShBIInterfaceNone = 0,

// Contract related functions ShBIInterfacePopulateContract = 1, ShBIInterfacePurgeContract = 2,

// Invoice related functions ShBIInterfacePopulateInvoice = 101, ShBIInterfacePurgeInvoice = 102,
ShBIInterfaceInvoiceClosedDate = 103,

// Debit related functions ShBIInterfacePopulateDebit = 201, ShBIInterfacePurgeDebit = 202,
ShBIInterfaceDebitClosedDate = 203,

// Payment related functions ShBIInterfacePopulatePayment = 301, ShBIInterfacePurgePayment =
302, ShBIInterfacePaymentClosedDate = 303,

// Credit related functions ShBIInterfacePopulateCredit = 401, ShBIInterfacePurgeCredit = 402,
ShBIInterfaceCreditClosedDate = 403,

// Ad order related functions ShBIInterfacePopulateAdOrder = 501, ShBIInterfacePurgeAdOrder =
502, ShBIInterfaceAdOrderQueueStatus = 503, ShBIInterfaceAdOrderInvoicedAlready = 504,
ShBIInterfaceAdOrderEditorialInsert = 505, ShBIInterfaceAdOrderPrimarySalesRep = 506,

// Miscellaneous functions ShBIInterfacePopulateCustomer = 901,
ShBIInterfaceStatementNumber = 902, ShBIInterfaceSyncAccountingPeriods = 903,

ShBIInterfaceLast = 903

} ShBIInterfaceEnumType;
```

Function	Function Code
<b>Contract related functions</b>	
ShBIInterfacePopulateContract	1
ShBIInterfacePurgeContract	2
<b>Invoice related functions</b>	
ShBIInterfacePopulateInvoice	101
ShBIInterfacePurgeInvoice	102
ShBIInterfaceInvoiceClosedDate	103
<b>Debit related functions</b>	
ShBIInterfacePopulateDebit	201
ShBIInterfacePurgeDebit	202
ShBIInterfaceDebitClosedDate	203
<b>Payment related functions</b>	
ShBIInterfacePopulatePayment	301
ShBIInterfacePurgePayment	302

Function	Function Code
ShBIInterfacePaymentClosedDate	303
<b>Credit related functions</b>	
ShBIInterfacePopulateCredit	401
ShBIInterfacePurgeCredit	402
ShBIInterfaceCreditClosedDate	403
<b>Ad order related functions</b>	
ShBIInterfacePopulateAdOrder	501
ShBIInterfacePurgeAdOrder	502
ShBIInterfaceAdOrderQueueStatus	503
ShBIInterfaceAdOrderInvoicedAlready	504
ShBIInterfaceAdOrderEditorialInsert	505
ShBIInterfaceAdOrderPrimarySalesRep	506
<b>Miscellaneous functions</b>	
ShBIInterfacePopulateCustomer	901
ShBIInterfaceStatementNumber	902
ShBIInterfaceSyncAccountingPeriods	903
ShBIInterfaceLast	90

## Application Ids

Application	Id
AdBookingAppId	1
ContactManagementAppId	2
GraphicsManagementAppId	3
LockAdminAppId	4
GalleyoutAppId	5
AdRaterAppId	6
ProductManagerAppId	7
AdOrderExportAppId	8
ScheduleSrvAppId	9
AdmarcCustomerExportAppId	10
AIMCustomerExportAppId	11
OfficePayMgrAppId	12
OutputAppId	13
AdOneExportAppId	14
Layout8000ExportId	15
CashReceiptsId	16
CustomerBalanceUtilityId	17
DisplayAdImporterId	18
InvoiceGeneratorId	19
RemoteDataAppId	20
TillForbidExtenderAppId	21
RemoteMaintAppId	22
AdOrderPurgeUtilityId	23
CustomerPurgeUtilityId	24
RemoteClientDataSenderAppId	25
CreditManagerUtilityId	26
ContractRenewUtilityId	27
DBTableUtilityId	28
WorkFlowUtilityId	29
SystemAdminId	30

Application	Id
AdbaseWebServerAppld	31
GEACCustomerExportId	32
PaymentDaemonId	33
GEACCreditImporterId	34
NovalInterfaceId	35
FinanceManagerId	36
GEACContractExportId	37
WebExporterId	38
GraphicPurgeUtilityId	39
ContractManagerAppld	40
BlindBoxReplyManagerId	41
LockBoxUtilityId	42
WriteOffUtilityId	43
NovaViaWarpInterfaceAppld	44
AuthorizeDotNetInterfaceAppld	45
CrainCustomerExportId	46
CrainContractExportId	47
AdBaseXMLImporterAppld	48
UserFieldsImportId	49
HighPlainsAdOrderExportId	50
HighPlainsCustomerExportId	51
GannettCustomerImporter	52
PlainDealerImportAppld	53
ExpireAdsId	54
PaymentPlusAXId	55 // Payment Plus interface
SAOEImportAppld	56
ICVerifyAXId	57
SuperChargeInterfaceId	58
SmartStreamAdOrderExportId	59
MactivePGLExportId	60

Application	Id
SmartStreamInterfaceId	61
PlainDealerAdOrderExportId	62
CampaignManagerAppId	63
ImportEditorialInformationAppId	64
PlainDealerCCInterfaceId	65
Annapolis8000ExportId	66
PlanPagExportId	67
SJCC8000ExportId	68
SystemTranslatorUtilityId	69
AdBaseWebServerId	70
AppSettingsManagerId	71
DataAnalysisReportId	72
DataAnalysisToolId	73
SpokaneCustomerExportId	74
WebDbManagerId	75
XMLWebExporterId	76
DataImportAppId	77
SpokaneCustomerImportId	78
AdjustmentManagerAppId	79
PlanPagImportId	100
AdQuoterId	101
ExternalLoginAppId	102
InvoiceAutoPayAppId	103
CustomerExportId	104
CreditBallImportId	105
GraphicsExporterAppId	106
RefundCheckExporterAppId	107
TellanAXAppId	108
GLExporterAppId	109
VerisignPayflowAppId	110

Application	Id
AutoLaunchToolAppld	111
CustomerImportAppld	112
AdTrackingExportAppld	113
TimesPicayune8000ExporterId	114
AdbaseXMLExporterAppld	115
AmosCustomerExportAppld	116
SanJoseGLEExportAppld	117
ContraCostaGLEExportAppld	118
NewarkOrderExporterAppld	119
SJCCGLEExportAppld	120
SystemNodeMonitorAppld	121
AdbaseSystemNodeManagerAppld	122
AdbaseToPMPAdtrackAppld	123
AdjustmentsMonitorAppld	124
AdbasePurgeUtilityAppld	125
RevenueExportAppld	126
BIPopulatorAppld	127
EmailAlertManagerId	128
GatewayPaymentsMgrAppld	129
InCollectionsMarkerId	130
BalanceUtilityId	131
IpixManagerId	132
BIImporterAppld	133
Fayetteville8000ExportId	134
RangerExportAppld	135
USPSConversionUtilityAppld	136
StatementNumberToolAppld	137
TimesPicayuneUSGExportAppld	138
TimesPicayunePPBillingExportAppld	139
MactivePlannerAppld	140

Application	Id
AdbaseXMLDistributorAppId	141
MactiveLoginAuditTrailId	142
XMLCustomerTransportId	143
CallFXExportAppId	144
BIUtilityAppId	145
BIInterfaceAppId	146
KubraInterfaceAppId	147
ScheduleAdminAppId	148
MGDSIProjectedCountsAppId	149
MGDSIFinalCountsAppId	150
AdPayManagerId	151
MobileAdPackPreprintId	152
BonusSpendingUtilityAppId	153
MactiveImageManagerId	154
MactiveImageUtilityAppId	155
DorfmanInterfaceAppId	156
Newark8000ExportId	157
VVPaymentInterfaceAppId	158
LiveProcessorInterfaceAppId	159
BookedSizeUpdaterAppId	160
MediaLinkAppId	161
PlaceholderOrderAppId	162
DocumentEpxorterAppId	163
StandardReportsAppId	164
CyberSourceAppId	165
AdbaseToYahooHotJobsId	166
CustomerStatusUtilityId	167
SDEPayInterfaceId	168
ContractImportAppId	169
GraphicsImporterAppId	170

Application	Id
PaywareAppld	171
DirectDebitToolAppld	172
AdbaseARAppld	173
InterfaceDateUtilityId	174
PreprintCountToolAppld	175
DJBizServicesInterfaceAppld	176
CreditCardUtility	193
EPSImporterAppld	194
PaymentCutoffUtilityAppld	195
SalesforceUpdaterAppld	196
TagOrderToolAppld	197
BatchCheckerAppld	198
DJBizPayToolAppld	199
OASExporterAppld	200
PDFLinkerAppld	201
AdbasePrintAuditAppld	202
PersonalFinanceManagerAppld	203
WebInventoryManagerAppld	204
PaymentTokenManagerAppld	205
BraintreePaymentAppld	206
GraphicsUpdaterAppld	207
XmlImporterFolderServerAppld	208
SSPCustomerExportAppld	209
OneViewAppld	210
EdgilEccOAppld	211
ElavonProtoBaseAppld	212
AutoPlannerAppld	213
BcContentMgmtSubsystemId	214
BcSpaceMgmtSubsystemId	215
TrafficManagerAppld	216



Application	Id
SCANDARFinanceChargerAppId	217
OneViewSelfServiceAppId	218
FTNIAppId	219
PayFlowTokenAppId	220
SPHWorldPayAppId	221
EPXAppId	222
EdgilPayWayAppId	223
FDHelperAppId	224
Newsday8000ExportId	225
AdServingAppId	226
PreSalesAppId	227
MSSExportAppId	228
AdSafeExportId	229
BroadcastFormatImporterId	230
BroadcastPostProductionId	231
BroadcastFeedbackImporterId	232
GLRemapperAppId	233
ASISStatsUtilityAppId	234
IntegrationServicesAppId	235
RelationUpdateInterfaceId	236
MonerisAppId	237
EPSFPurgerAppId	238
GoalToolAppId	239
ChasePaymentTechAppId	240
ISNPDFXMLExporterAppId	241
ImportEditorialInfoAppId	242
PrintQueueToolAppId	243
ReportToolAppId	244
FirstDataAppId	245
FieldedDataModUtilAppId	246

Application	Id
DJContractAccrualExportAppld	247
FinanceChargerAppld	248

Function	Function Code
<b>Contract related functions</b>	
ShBIInterfacePopulateContract	1