

Contents

Data Dictionary	3
All Tables	3
Accounts Master	9
Product Master.....	11
Department/Tags1	12
Cost Center/Tags2.....	13
Warehouse/Tags3.....	14
Site/Tags4.....	15
Employee/Tags5.....	15
Machine/Tags6.....	16
Region/Tags7.....	16
Fixed Asset/Tags8	17
Bins.....	18
Sample queries on Accounts Table.....	18
Data Table.....	20
Data Header	23
Inventory	24
Fixed Assets	24
Point of Sales (POS)	27
Human Resources Management (HRM).....	38
Payroll Preferences	38
Employee Master	39
Monthly Payroll.....	43
Leaves Application	47
Advances.....	48
Arrears.....	49
Sector	50
Shifts	50
Time-Sheet.....	51
Vacation	53
Training.....	55
Achievements	55
Payroll External	56
Recruitment.....	62
API - Introduction.....	64
COM Registration.....	64
CLASS LIST	65
Company.ocx	66
Field List.....	66
Sample Code.....	66
Option.ocx	67
Field List.....	67
Sample Codes	67
FMaster.....	68
Methods	69
Direct Methods	70
Field List.....	71
Sample Code.....	71
FTransaction	72
Methods	72
Direct Methods	74
Field List.....	74

Sample Code.....	75
FReport.....	76
Methods	76
Direct Methods	77
Fields	78
Sample Code.....	79
FMasterReport	80
Methods	80
Field List.....	80
Sample Code.....	81
FLinks	81
Methods	81
Field List.....	83
Sample Code.....	83
References	84
Methods	84
Field List.....	85
Sample Code.....	85
FUnits	86
Methods	86
Field List.....	86
Sample Code.....	87
FCurrencies	87
Methods	87
Sample Code.....	88
FRateMaster	89
Methods	89
Direct Methods	90
Sample Code.....	90
FFixedAssets	91
Methods	91
Field List.....	91
Sample Code.....	92
FPayroll.....	93
Methods	93
Field List.....	94
Sample.....	94
LOGINS.....	96
Methods	96
Sample Codes	96
FRemoteExec	97
Methods	97
Sample Code.....	98

Data Dictionary

All Tables

List of tables used in FocusRT					
T y p e	Object Name	Module	Objective	Documents that update the table	Linked Tables
T	addvl	Fixed Asset	Stores Additional Values of Fixed Assets	\Fixed Asset\Change Asset Value	astrg
T	adspl	Fixed Asset	Stores Disposal of Fixed Assets	\Fixed Assets\Disposal of Asset	astrg
T	altprod	Manufacturing	Stores Alternate Product given in the BOM	\Transaction\Manufacturing\Set Alternate Product in Bill of Mater	Bname
T	amntn	Fixed Asset	Stores Maintenance Details of Assets	\Fixed Asset\Maintenance	astrg
T	arepr	Fixed Asset	Stores the repair details of Assets	\Fixed Asset\Repairs	astrg
T	astrg	Fixed Asset	It's Asset Register, stores all details of the asset including Value, Depreciation Method, Depreciation posting GL accounts etc	\Fixed Asset\Fixed Asset Tree	mr009
T	atrfr	Fixed Asset	Stores transfer of Asset from one location to another	\Fixed Asset\Transfer of Asset	mr009, astrg
T	ausag	Fixed Asset	Stores the usage details of Asset	\Fixed Asset\Assets Usage	mr009, astrg
T	batch	Inventory	Stores the inventory Batch details (With expiry and manufacturing dates)	\Transaction\All inventory Documents	indta
T	bins	Inventory	Stores selected bin position of each product	\Transaction\All inventory Documents	data
T	bname	Manufacturing	Stores the Production Batch Details	\Transaction\Manufacturing\Add\Edit Batch	mr001, bomin
T	Bomdt	Manufacturing	Stores the	\Transaction\Manufacturing	mr001,

			input/output details of Bill of Material	\Bill of Material	bomin
T	BomIn	Manufacturing	Stores the header details of Bill of Material	\Transaction\Manufacturing\Bill of Material	bomdt
T	budgs	Financials	Stores account Budgets	\Company\Create Master\Accounts Tree	mr000, mr002
T	Curnc	Financials	Stores the currency Rates	\company\Currency Exchange Rates	
T	custflds	All	Stores the details of extra field link tables	\Company\VoucherWizard\	
T	dailyforecast	MRP	stores the MRP daily production plan	\Company\Production planning\Daily Production Plan	mr001
T	data	All	Stores the information of data related to financial, Inventory, Orders, Production	\Transaction\All Documents	Explained in Separate Tab
T	deleted	All	keeps the deleted records history for audit trail		
T	deprn	Financials	Stores the depreciation details entered from accounts tree (Those not using Fixed Asset Module)	\Company\Create Master\Accounts Tree	mr000
T	deptappr	Financials	Stores the appropriation of account/departm ent percentages	\Company\Create Master\Accounts Tree	mr000, mr002
T	detailedanalysis	MRP	Stores Raw Material Purchase Requirement from Production Planning	\Transaction\Purchase Planning\Monthly Raw Material Analysis	mr001
T	detailedanalysis body	MRP	Stores the detail part of raw material requirement	\Transaction\Purchase Planning\Monthly Raw Material Analysis	
T	errorslog	All	Stores the errors during reposting/reindex ing		
T	header	All	Stores the header part of all documents like login name, no. of entries, authorization...	\Transaction\All Documents	data

T	header2	All	Stores inventory related header portion related to production and stock transfers	\Transaction\All inventory Documents	data
T	header2deleted	All	stores the audit of deleted header 2 table		
T	headerdeleted	All	Stores the records of deleted header table		
T	ibals	Inventory, Orders, Manufacturing	stores inventory balances by date and location	\Transaction\All inventory Documents	mr001, mr004 (Depends on default tag for inventory)
T	indta	Inventory, Orders, Manufacturing	Stores all inventory related detailed portion like Quantity, Gross and extra screen columns	\Transaction\All inventory Documents	data, batch
T	indtadeleted	Inventory	Stores deleted records of indta table		
T	LastYearBatchIssues	Manufacturing	Stores carry forward semi-finished batches and its cost from previous year	\Transaction\Finalization of Accounts\Close Accounts	
	Incry	All/Optional	Stores previous years pending links	\Transaction\Finalization of Accounts\Close Accounts	
T	logbk	All	Stores login information and menu access of each user		
T	MacBr	Manufacturing	Machine Breakdown		
	MaintenanceDefBody	Fixed Asset	Stores the detail information of Asset Maintenance	\FixedAsset\Maintenance\Define Maintenance	
	MaintenanceDefHeader	Fixed Asset	Stores the header portion of Asset Maintenance	\FixedAsset\Maintenance\Define Maintenance	
T	monthlyforecastref	MRP	Stores the monthly Production forecast	\Transaction\Production Planning\Monthly Production Plan	
T	mr000	Financials	Accounts Master	\Company\Create Master\Accounts Tree	u0000
T	mr001	Inventory,	Product Master	\Company\Create	u0001,

		Orders, Manufacturing		Master\Products Tree	srate, prdet
T	mr002	All/Optional	Department (Tag 1)	\Company\Create Master\Department	
T	mr003	All/Optional	Cost Center (Tag 2)	\Company\Create Master\Cost Center	
T	mr004	All/Optional	Warehouse (Tag 3)	\Company\Create Master\Warehouse	
T	mr005	All/Optional	Site (Tag 4)	\Company\Create Master\Site	
T	mr006	HR & Payroll	Employee Master	\Payroll & HR\Employee Details\Create Employee	
T	mr007	All/Optional	Region	\Company\Create Master\Region	
T	mr008	All/Optional	Machine	\Manufacturing\Create Machine	
T	mr009	Fixed Asset	Fixed Assets	\Fixed Asset\Fixed Asset Tree	
T	mr00a	Inventory	Bins	\Company\Create Master\Bins	
	mrp	MRP	Production planning based on Order	\Transaction\Production Planning\Production Planner	
	mrpres	MRP	Reservation through MRP	\Transaction\Production Planning\Production Planner	
	NonPrvntMntBody	Fixed Asset	Details of non preventive Maintenance	\Fixed Asset\Maintenance\Repairs & Maintenance	
	NonPrvntMntHeader	Fixed Asset	Header part of non preventive maintenance	\Fixed Asset\Maintenance\Repairs & Maintenance	
T	prdet	Financials, Inventory	Product Details related to inventory account, cost of sale etc	\Company\Create Masters\Products Tree	mr001, mr000
	procskip	Manufacturing	Process skip document	\Transaction\Manufacturing\skip production process	bname
T	ProdBatch	Manufacturing	Allocation of Machine to batch	\Transaction\Manufacturing\reserve Machine/Manpower	
T	ProdBatchIssues	Manufacturing	Actual usage of Machine for each batch	\Transaction\Manufacturing\Batch-wise Machine/Manpower usage	
T	prodforecastbody	MRP	Sales Forecast related to MRP, details	\Transaction\Production Planning\Sales Forecast	
T	prodforecastheader	MRP	Sales Forecast related to MRP, Header	\Transaction\Production Planning\Sales Forecast	
T	prodforecastprod	MRP	Product details related to Sales Forecast	\Transaction\Production Planning\Sales Forecast	
T	ProdPlan	MRP	Sales Forecast	\Transaction\Production	

			when converted to production plan	Planning\Annual Production Plan	
T	provbody	Financials	Stores Provisional Entry from Final Reports	\Financial Accounting\Final Accounts\Trail Balance	mr000
T	provheader	Financials	Stores the header portion of Provisional Entries	\Financial Accounting\Final Accounts\Trail Balance	provbody
T	prttarget	MRP	Production Actual Vs Target		
T	prttargetbackup	MRP	Backup of prttarget		
T	qbudg	Inventory	Quantity Budgets		
T	Refrn	Accounts Receivable/Payable	Pending/Adjusted Invoices Reference Table	\Transaction\All Documents	data
T	restk	Inventory	Reserved Quantity Balances		
T	rma	Inventory	Returned Merchandize Authorization (Serial Numbers of products)	\Transaction\All inventory Documents	data
T	setitems	Inventory	Record of Sets created in items Tree	\Company\Create Master\Products Tree	mr001
T	setnames	inventory	Header portion of Setitems	\Company\Create Master\Products Tree	Setitems
T	srate	Inventory, Orders, Manufacturing	Purchase/Sales Rate of each product	\Company\Create Master\Products Tree	mr001, mr001
T	u0000	Masters	Extra Fields data defined for Accounts		mr000
T	u0001	Masters	Extra Fields data defined for Products		mr001
T	u0002	Masters	Extra Fields data defined for Department(Tag1)		
T	u0003	Masters	Extra Fields data defined for Cost Center (tag2)		
T	u0004	Masters	Extra Fields data defined for Warehouse (Tag3)		
T	u0005	Masters	Extra Fields data defined for Site (Tag4)		
T	u0006	Masters	Extra Fields data		

			defined for Employees		
T	u0007	Masters	Extra Fields data defined for Region		
T	u0008	Masters	Extra Fields data defined for Machine		
T	u0009	Masters	Extra Fields data defined for Fixed Assets		
T	u000a	Masters	Extra Fields data defined for Bins		
T	u0010	Financials	Default Extra Field Layout for Receipt/Payment Voucher		
T	u0011	Financials	Default Extra Field Layout for Journals		
T	u0012	Finance/Inventory	Default Extra Field layout for Sales related documents		
T	u0013	Finance/Inventory	default Extra Field Layout for Purchase Related documents		
T	u0014	Finance/Inventory	default extra field layout for Inventory related documents		
T	U???		As you keep defining extra layouts for different documents, you will get a new U file. For actual link you have to refer to custflds table		
T	units	Inventory	Unit defined for each product	\Company\Create Master\Products Tree	mr001
T	vtypes	All	Voucher Type definition linked to data		data
V	allaccts	Financials	All Financial Transactions from data table which is used in Report Designer set for all accounting transactions		
V	allprods	Inventory	All Inventory Transactions affecting inventory		
V	datafa	Financials	All financial Transactions from data table with Debit and Credit in one row		
V	pendingbatches	Inventory	Gives Inventory batches and its balance qty		data, indta, mr001
V	pendingrefs	Financials	Gives pending bills and its amount		data, mr000
V	pendingrefsdep	Financials	returns pending		

	wise		bills/invoices by department		
F	balanceon				
F	ConvertFocusDate	All	converts the integer value of date to date in yyyyymmdd format		
F	ConvertFocusDateDisplay	All	Converts and displays in dd-mm-yyyy format		
F	ConvertFocusDateHijri	All	Converts Focus integer value in hijri to hijri/Farsi Date		

Accounts Master

Table Name	Columns	Type	Description
mr000	MasterId	Int	Primary Key, mapped to all other tables in the database
	Sequence	Int	Stores the sequence of its position in the chart of Accounts (Accounts Tree)
	Name	Varchar(60)	Stores the Name of the account
	Code	Varchar(55)	Stores the Alias (alternate Name), name in other language.
	Code2	Varchar(30)	Stores the Account Code.
	Limit	Float	Stores the balance limit of the customer
	L2	Int	Stores the parent MasterId. -1 denotes no parent/0 level master
	Type	Type	Type stores the multiple information in bits: account type, group/leaf and the Group level in the tree. See Sample query for Master
	Attribute	Attribute	Attribute stores the properties of account: Active/Inactive, Display detail/Hide Detail, and Annual/Monthly Ledger. Sample query for Master
	Eoff	Int	This field is mapped to ExtraId of u0000 table which stores the extra information created using customized fields
	Doff	Int	Not in use
	CreditDays	smallint	Stores the credit days for customer/vendor accounts
	Date_	int	Stores the date of creation/last edit in integer form. See sample query to convert date_ from integer to Date function
	Time_	binary(3)	Stores time of creation/last edit. Time is stored in binary format. For conversion into Hours, Minutes and Seconds check sample query on

			time conversion
	Limit2	Float	Not in use in accounts master

Related Tables where MasterID of mr000 is linked

Table Name	Field Name	Description
data	Code	
data	BookNo	
Header2	BodyAccts (N)	Multiple depending on the columns created in inventory related screens
	HeaderAccts(N)	Multiple fields depending on the footer fields created in the inventory type screens
Astrg	AssetAcCode	Stores the MasterId of Asset Account in the Fixed Asset Master
	DepreCode	Stores the MasterID of Accumulated Depreciation Account in the Fixed Asset Master
	DeprePLCode	Stores the MasterID of Depreciation Account in the Fixed Asset Master
	SalesAccount	Stores the MasterId of Sales Account in the Fixed Asset Master
Budgs	Party	Stores the MasterId of account for which budget is defined in budget screen
PosHeader	SalesCode	Stores the MasterId of Sales Account in Point of Sale transaction
	CustCode	Stores the MasterId of the Customer/Cash Account in Point of Sale Transaction
	CustCode2	Stores the MasterID of the credit Customer in Point of sale payment details
PosPref	SalesAccountID	Stores the MasterID Sales Account
	CashAccountID	Stores the MasterID Cash Account
	DebitAccountID	Stores the MasterID Customer/Debit Account
	CreditAccountID	Stores the MasterID Sales on Credit Account
	DrAct_CrSale	Stores the MasterID Customer sold on Credit Account
	CouponID	Stores the MasterID Coupon Account
	SuspenseID	Stores the MasterID Suspense Account
	SpecialDiscID	Stores the MasterID Special Discount Account
	TaxID	Stores the MasterID Tax Account
	DiscountID	Stores the MasterID Discount Account
Prdet	Cogs	Stores the MasterId of account defined as cost of Sale account in product master
	SalesAc	Stores the MasterId of account defined as Sale account in product master
	PurchasesAc	Stores the MasterId of account defined as Stock account in product master
	Supplier(N)	Stores the MasterId of account defined as supplier account in product master
Refrn	Code	Stores the MasterId of the customer/vendor whose bill is adjusted
Srate	Pty	Stores the MasterID of the customer/vendor Account
TdsDetails	IAcc	Stores the Masterid of TDS account
Tdsrates	Iacc	Stores the Masterid of Accounts whose TDS rate is

		defined
u0000	ExtraId	Stores the Eoff of mr000 table. This table stores the extra fields data of the accounts master

Product Master

Table Name	Columns	Type	Description
mr001	MasterId	Int	Primary Key, mapped to all other tables in the database
	Sequence	Int	Stores the sequence of its position in the chart of Products (Products Tree)
	Name	Varchar(60)	Stores the Name of the Product
	Code	Varchar(55)	Stores the Alias (alternate Name), name in other language.
	Code2	Varchar(30)	Stores the Item/Product Code.
	Limit	Float	Stores the Reorder Level of the Product
	L2	Int	Stores the parent MasterId. -1 denotes no parent/0 level master
	Type	Type	Type stores the multiple information in bits: Product type, group/leaf and the Group level in the tree. See Sample query for Master
	Attribute	Attribute	Attribute stores the properties of Product: Active/Inactive, Do Not Use Batch, Use Quantity Breakup. Sample query for Master
	Eoff	Int	This field is mapped to ExtraId of u0001 table which stores the extra information created using customized fields
	Doff	Int	Not in use
	CreditDays	smallint	Not in Use
	Date_	int	Stores the date of creation/last edit in integer form. See sample query to convert date_ from integer to Date function
	Time_	binary(3)	Stores time of creation/last edit. Time is stored in binary format. For conversion into Hours, Minutes and Seconds check sample query on time conversion
	Limit2	Float	Stores the Bin Capacity, The figure here is multiplied with Bin Capacity to arrive at total quantity that can be stored in a bin

Related Tables where MasterID of mr001 is linked		
Table Name	Field Name	Description
AltProd	OldProd	Stores the MasterID of Old product
	NewProd	Stores the MasterID of New product
bname	Product	Stores the MasterId of the product for which production batch is created
BomDt	Product	Stores the MasterId of the product entered as input/output in the BOM screen
data	ProductCod	Stores the MasterId of the product entered in the

	e	transaction
ibals	Code	Stores the MasterId of the products whose issue/receipt summary is maintained for stock balance
ItemsReturn	ProdCode	Stores the MasterId of the products return at point of sale
Line	ProdCode	Stores the MasterId of the Products sold through point of sale screen
mrporder	Product	Stores the MasterId of products selected in MRP screen
Prdet	MasterId	Stores the MasterId of the product and the details entered in other details tab
ProdForecastProd	Product	Stores the MasterId of the product which is used in the sales forecast
ProdPlan	Product	Stores the MasterId of the product picked from BOM in the Annual Purchase Plan
ProductReorder	iProductid	Stores the MasterId of the product picked for reorder in the Monthly Purchase Plan
PrTarget	Code	Stores the MasterID of the product picked in forecast and planned production
qbudg	Product	Stores the masterId of the product used in defining quantity budget
ReqBody	Product	Stores the MasterId of the product used in Requisition
Restk	Code	Stores the MasterId of product. This table stores the details of reserved/released stock balances
SetItems	Product	Stores the MasterId of the items used in the set
u0001	Extraid	Stores the Eoff of product (mr001) table for extra information of the product
Units	MasterId	Stores the Masterid of the product whose units are defined in units table

Department/Tags1

Table Name	Columns	Type	Description
mr002	MasterId	Int	Primary Key, mapped to all other tables in the database
	Sequence	Int	Stores the sequence of its position in the chart of Department (Tags0 Tree)
	Name	Varchar(60)	Stores the Name of the Department/Tags0
	Code	Varchar(55)	Stores the Alias (alternate Name), name in other language.
	Code2	Varchar(30)	Stores the Department/Tags0 Code.
	Limit	Float	Not in use
	L2	Int	Not in use
	Type	Type	Not in use
	Attribute	Attribute	Not in use
	Eoff	Int	This field is mapped to ExtraId of u0002 table which stores the extra information created using customized fields
	Doff	Int	Not in use
	CreditDays	smallint	Not in use

	Date_	int	Stores the date of creation/last edit in integer form. See sample query to convert date from integer to Date function
	Time_	binary(3)	Stores time of creation/last edit. Time is stored in binary format. For conversion into Hours, Minutes and Seconds check sample query on time conversion
	Limit2	Float	Not in use

Related Tables where MasterID of Department (mr002) is linked		
Table Name	Field Name	Description
astrg	Location	Stores the masterid of department selected as location in the Fixed Assets Master only if default tag for fixed asset is set as Department.
atrfr	FromLocation	Stores the masterid of department selected in the assets transfer only if default tag for fixed asset is set as Department.
	ToLocation	Stores the masterid of department selected in the asset transfer only if default tag for fixed asset is set as Department.
budgs	dep	Stores the MasterId of the department selected in the account budget only if default financial tag is selected as department
data	Tags0	Stores the MasterId of the department selected in the transactions. Stores 0 for Vouchers that do not use department
deptappr	Dep	Stores the Masterid of the department selected in the Department appropriation in accounts tree only if department is used as default tag for FA
PosHeader	CounterCode	Stores the masterid of the department if department is used as counter in Point of Sale
ProdForecastheader	Dept	Stores the MasterId of the department for which sales forecast is entered.
Refrn	Dep	Stores the MasterID of the department to which invoice/receipt belongs to in bill adjustment for customer/vendor
u0002	ExtraId	Stores the Eoff of mr002 table.

Cost Center/Tags2

Table Name	Columns	Type	Description
mr003	MasterId	Int	Primary Key, mapped to all other tables in the database
	Sequence	Int	Stores the sequence of its position in the chart of Cost Center(Tags1 Tree)

	Name	Varchar(60)	Stores the Name of the Cost Center/Tags1
	Code	Varchar(55)	Stores the Alias (alternate Name), name in other language.
	Code2	Varchar(30)	Stores the Cost Center/Tags1 Code.
	Limit	Float	Not in use
	L2	Int	Not in use
	Type	Type	Not in use
	Attribute	Attribute	Not in use
	Eoff	Int	This field is mapped to ExtraId of u0003 table which stores the extra information created using customized fields
	Doff	Int	Not in use
	CreditDays	smallint	Not in use
	Date_	int	Stores the date of creation/last edit in integer form. See sample query to convert date_ from integer to Date function
	Time_	binary(3)	Stores time of creation/last edit. Time is stored in binary format. For conversion into Hours, Minutes and Seconds check sample query on time conversion
	Limit2	Float	Not in use

Warehouse/Tags3

Table Name	Columns	Type	Description
mr004	MasterId	Int	Primary Key, mapped to all other tables in the database
	Sequence	Int	Stores the sequence of its position in the chart of Department (Tags0 Tree)
	Name	Varchar(60)	Stores the Name of the Warehouse/Tags2
	Code	Varchar(55)	Stores the Alias (alternate Name), name in other language.
	Code2	Varchar(30)	Stores the Warehouse/Tags2 Code.
	Limit	Float	Not in use
	L2	Int	Not in use
	Type	Type	Not in use
	Attribute	Attribute	Not in use
	Eoff	Int	This field is mapped to ExtraId of u0004 table which stores the extra information created using customized fields
	Doff	Int	Not in use
	CreditDays	smallint	Not in use
	Date_	int	Stores the date of creation/last edit in integer form. See sample query to convert date_ from integer to Date function
	Time_	binary(3)	Stores time of creation/last edit. Time is stored in binary format. For conversion into Hours, Minutes and Seconds check sample query on time conversion

	Limit2	Float	Not in use
--	--------	-------	------------

Site/Tags4

Table Name	Columns	Type	Description
mr005	MasterId	Int	Primary Key, mapped to all other tables in the database
	Sequence	Int	Stores the sequence of its position in the chart of Site (Tags3 Tree)
	Name	Varchar(60)	Stores the Name of the Site/Tags3
	Code	Varchar(55)	Stores the Alias (alternate Name), name in other language.
	Code2	Varchar(30)	Stores the Site/Tags3 Code.
	Limit	Float	Not in use
	L2	Int	Not in use
	Type	Type	Not in use
	Attribute	Attribute	Not in use
	Eoff	Int	This field is mapped to ExtraId of u0005 table which stores the extra information created using customized fields
	Doff	Int	Not in use
	CreditDays	smallint	Not in use
	Date_	int	Stores the date of creation/last edit in integer form. See sample query to convert date_ from integer to Date function
	Time_	binary(3)	Stores time of creation/last edit. Time is stored in binary format. For conversion into Hours, Minutes and Seconds check sample query on time conversion
	Limit2	Float	Not in use

Employee/Tags5

Table Name	Columns	Type	Description
mr006	MasterId	Int	Primary Key, mapped to all other tables in the database
	Sequence	Int	Stores the sequence of its position in the chart of Employee Tree
	Name	Varchar(60)	Stores the Name of the Employee
	Code	Varchar(55)	Stores the Alias (alternate Name), name in other language.
	Code2	Varchar(30)	Stores the Employee Code.
	Limit	Float	Not in use
	L2	Int	Not in use
	Type	Type	Not in use
	Attribute	Attribute	Not in use
	Eoff	Int	This field is mapped to ExtraId of u0006 table which stores the extra information

			created using customized fields
	Doff	Int	Not in use
	CreditDays	smallint	Not in use
	Date_	int	Stores the date of creation/last edit in integer form. See sample query to convert date_ from integer to Date function
	Time_	binary(3)	Stores time of creation/last edit. Time is stored in binary format. For conversion into Hours, Minutes and Seconds check sample query on time conversion
	Limit2	Float	Not in use

Machine/Tags6

Table Name	Columns	Type	Description
mr007	MasterId	Int	Primary Key, mapped to all other tables in the database
	Sequence	Int	Stores the sequence of its position in the chart of Machine Tree
	Name	Varchar(60)	Stores the Name of the Machine
	Code	Varchar(55)	Stores the Alias (alternate Name), name in other language.
	Code2	Varchar(30)	Stores the Machine Code.
	Limit	Float	Not in use
	L2	Int	Not in use
	Type	Type	Not in use
	Attribute	Attribute	Not in use
	Eoff	Int	This field is mapped to ExtraId of u0007 table which stores the extra information created using customized fields
	Doff	Int	Not in use
	CreditDays	smallint	Not in use
	Date_	int	Stores the date of creation/last edit in integer form. See sample query to convert date_ from integer to Date function
	Time_	binary(3)	Stores time of creation/last edit. Time is stored in binary format. For conversion into Hours, Minutes and Seconds check sample query on time conversion
	Limit2	Float	Not in use

Region/Tags7

Table Name	Columns	Type	Description
------------	---------	------	-------------

mr008	MasterId	Int	Primary Key, mapped to all other tables in the database
	Sequence	Int	Stores the sequence of its position in the chart of Region Tree
	Name	Varchar(60)	Stores the Name of the Region
	Code	Varchar(55)	Stores the Alias (alternate Name), name in other language.
	Code2	Varchar(30)	Stores the Region Code.
	Limit	Float	Not in use
	L2	Int	Not in use
	Type	Type	Not in use
	Attribute	Attribute	Not in use
	Eoff	Int	This field is mapped to ExtraId of u0008 table which stores the extra information created using customized fields
	Doff	Int	Not in use
	CreditDays	smallint	Not in use
	Date_	int	Stores the date of creation/last edit in integer form. See sample query to convert date_ from integer to Date function
	Time_	binary(3)	Stores time of creation/last edit. Time is stored in binary format. For conversion into Hours, Minutes and Seconds check sample query on time conversion
	Limit2	Float	Not in use

Fixed Asset/Tags8

mr009	MasterId	Int	Primary Key, mapped to all other tables in the database
	Sequence	Int	Stores the sequence of its position in the chart of Fixed Assets Tree
	Name	Varchar(60)	Stores the Name of the Fixed Assets
	Code	Varchar(55)	Stores the Alias (alternate Name), name in other language.
	Code2	Varchar(30)	Stores the Fixed Asset Code.
	Limit	Float	Not in use
	L2	Int	Not in use
	Type	Type	Not in use
	Attribute	Attribute	Not in use
	Eoff	Int	This field is mapped to ExtraId of u0009 table which stores the extra information created using customized fields
	Doff	Int	Not in use
	CreditDays	smallint	Not in use
	Date_	int	Stores the date of creation/last edit in integer form. See sample query to convert date_ from integer to Date function
	Time_	binary(3)	Stores time of creation/last edit. Time is stored in binary format. For conversion into Hours, Minutes and Seconds check sample query on time conversion

	Limit2	Float	Not in use
--	--------	-------	------------

Bins

Table Name	Columns	Type	Description
mr00a	MasterId	Int	Primary Key, mapped to all other tables in the database
	Sequence	Int	Stores the sequence of its position in the chart of Bin Tree
	Name	Varchar(60)	Stores the Name of the bins
	Code	Varchar(55)	Stores the Alias (alternate Name), name in other language.
	Code2	Varchar(30)	Stores the bin Code.
	Limit	Float	Stores the overall capacity of the bin
	L2	Int	Not in use
	Type	Type	Not in use
	Attribute	Attribute	Not in use
	Eoff	Int	This field is mapped to ExtraId of u000a table which stores the extra information created using customized fields
	Doff	Int	Not in use
	CreditDays	smallint	Not in use
	Date_	int	Stores the date of creation/last edit in integer form. See sample query to convert date_ from integer to Date function
	Time_	binary(3)	Stores time of creation/last edit. Time is stored in binary format. For conversion into Hours, Minutes and Seconds check sample query on time conversion
	Limit2	Float	Not in use

Sample queries on Accounts Table

To get Account Type

```

select Name,case When Type&0x0f=0 Then 'CASH' Else
                case When Type&0x0f=1 Then 'Bank' Else
                case When Type&0x0f=2 Then 'Employee' ELSE

                case When Type&0x0f=3 Then 'Cust Vendor' ELSE
                case When Type&0x0f=4 Then 'Asset liability' ELSE
                case When Type&0x0f=5 Then 'Sales' ELSE
                case When Type&0x0f=6 Then 'Purchase' ELSE
                case When Type&0x0f=7 Then 'Income Expense' ELSE
                case When Type&0x0f=8 Then 'Trading' Else 'None'
                END
                END
                END
                END
                END
                END

```

```

END
END
END
from mr000

```

To get Tree View

```

select (type/16&0x07) Level,
      Case When ((type/16&0x07)=0) then Name
        When (type/16&0x07)=1 then ' '+Name
        When (type/16&0x07)=2 then '  '+Name
        When (type/16&0x07)=3 then '   '+Name
        When (type/16&0x07)=4 then '    '+Name
        When (type/16&0x07)=5 then '     '+Name
        When (type/16&0x07)=6 then '      '+Name
        else ' '
      End Ac_Name,
      Case When ((type&0x80)=0) then 'Group'
        else 'Leaf'
      End Acc_Type
from mr000 order by sequence

```

To get Level of account in the tree

```

select type/16&0x07 Level, * from mr000

```

To find if Account is a leaf or Group

```

select type&0x80 Grp, * from mr000
(return 128 for leaf and 0 for group)

```

Note: Level and Group Query applies to all the master tables

To get Product Type

```

select Name, Case When Type&0x0f=0 Then 'Service' Else
      Case When Type&0x0f=1 Then 'Raw Material' Else
      Case When Type&0x0f=2 Then 'Intermediate Product' ELSE

      Case When Type&0x0f=3 Then 'Finished Good'
      END
      END
      END
      END
from mr001

```

To get Units Name from Inventory, Product Units tables

```

select m.Name,
      Case When cast(Unit as int)= 0 Then U.Units0 Else
      Case When cast(Unit as int)= 1 Then U.Units1 Else
      Case When cast(Unit as int)= 2 Then U.Units2 Else
      Case When cast(Unit as int)= 3 Then U.Units3

```

End
End
End
End,

i.Quantity from mr001 m join data d on m.MasterId=d.ProductCode join indta i on d.SalesOff=i.SalesId left join units u on d.ProductCode=u.MasterId

To Convert Date/Time in any Table from integer to date/Time format

The below query will give date in the dd-mm-yyyy format

```
select dbo.ConvertFocusDateDisplay(Date_), cast(Time_ as int)/(256*256)&0xff Hours,
cast(Time_ as int)/(256)&0xff Minutes, cast(Time_ as Int)&0xff seconds, Name, Code2
from mr000
```

To get date in ddmmyyyy format

```
select dbo.ConvertFocusDateDisplay(Date_), cast(Time_ as int)/(256*256)&0xff Hours,
cast(Time_ as int)/(256)&0xff Minutes, cast(Time_ as Int)&0xff seconds, Name, Code2
from mr000
```

Data Table

Field Name	Description	Sample Sql Query
Body Id	Stores unique Record Numbers, not linked to any other table	
Date_	Stores integer value of date, can be converted to date format using the built-in function given.	select dbo.ConvertFocusDateDisplay(Date_), * from data
Vouchertype	Stores the integer value of each voucher type defined in vtypes	select v.vt, d.* from data d, vtypes v where v.VoucherType=d.VoucherType
VoucherNo	Document Number of the transactions	
Code	Stores the MasterId of the Debit account	select m.Name, d.* from data d, mr000 m where m.MasterId=d.Code
BookNo	Stores the contra Account for the transaction	select m.Name, d.* from data d, mr000 m where m.MasterId=d.BookNo
ProductCode	Stores the MasterId from product Master	select m.Name, d.* from data d, mr001 m where m.MasterId=d.ProductCode
Tags0	Stores the 1st Tag (Department) if in use	select m.Name, d.* from data d, mr002 m where m.MasterId=d.Tags0
Tags1	Stores the 2nd Tag (Cost Center) if in use	select m.Name, d.* from data d, mr003 m where m.MasterId=d.Tags1
Tags1	Stores the 3rd Tag (Warhouse) if in use	select m.Name, d.* from data d, mr004 m where m.MasterId=d.Tags2

Tags1	Stores the 4th Tag (Site) if in use	select m.Name, d.* from data d, mr005 m where m.MasterId=d.Tags3
Amount1	Stores the Debit Amount, Transaction Gross Converted to Base currency	
Amount2	Stores the net value of the voucher or Credit Amount (if detailed Posting is On)	
OriginalAmount	Stores the Entered Value or Rate before conversion to Base Currency	
Flags	Stores multiple information in bits.	
	Financial Records only	select v.vt, d.* from data d join vtypes v on d.VoucherType=v.VoucherType where (d.Flags&0x400)=0x400
	Transactions updating Inventory Module Only (irrespective of whether updating FA or not)	select v.vt, d.* from data d join vtypes v on d.VoucherType=v.VoucherType where (d.Flags&0x200)=0x200
	Records updating Inventory and Financial Both	select v.vt, d.* from data d join vtypes v on d.VoucherType=v.VoucherType where (d.Flags&0x600)=0x600
	Records that update neither Inventory nor Financials	select v.vt, d.* from data d join vtypes v on d.VoucherType=v.VoucherType where (d.Flags&0x600)=0
	Stores the currency Id	select (Flags/2)&0x7f, * from data
	Stores the suspended document record	select * from data where (Flags&0x400000)=0x400000
	Stores the converted/unconverted flag for Memo/PDC	select * from data where (Flags&0x200000)=0x200000
	Stores Locked/Unlocked Flag	select * from data where (Flags&0x20)=0x20
	Stores Checked/Unchecked Flag	select * from data where (Flags&0x8000)=0x8000
	Stores closed/open Link	select * from data where (Flags&0x20000)=0x20000
	Stores Reserve Stock	select p.MasterId, sum(i.Quantity) reserve from indta i join data d on d.SalesOff=i.SalesId join mr001 p on p.MasterId=d.ProductCode
	Stores release Stock	where (d.Flags&0xC000)=0x4000 group by p.MasterId
	Stores the BRS Flag	select * from data where Vouchertype=4608 and data.flags&0x2000=0x2000
Due Date	Stores the Due date of the voucher in integer format.	
ExtraOff	Stores the body extra fields link with the extra field table. Consider here u0110 which is linked to Receipt/Payment, other u files can be found from custflds table	select u.RemarksYH, d.* from data d, u0110 u where u.ExtraId=d.Extraoff

HeaderOff	Stores the link to the header table that stores the login information of the document	select h.Login, ConvertFocusDate(h.Date_), d.* from header h, data d where h.HeaderId=d.HeaderOff
SalesOff	Stores the link between the inventory details like Quantity, Rate, Qty2, additional Columns	select i.Quantity, i.Rate, i.StockValue, d.* from data d,
BillwiseOff	Stores the link between Pending and adjusted references	create view newref as select d.VoucherNo, v.Vt, r.Amount, r.Code, d.BookNo, d.BillWiseOff, d.Date_, d.Tags0 d.Tags1, d.Tags2,r.Ref&0x7fffffff NewRef from data d, vtypes v, Refrn r whered.VoucherType=v.VoucherType and d.BillWiseOff=r.RefId and Ref<0
		Create view agref as select sum(r.Amount) Adjust, r.Ref from data d, vtypes v, Refrn r where d.VoucherType=v.VoucherType and d.BillWiseOff=r.RefId and Ref>0 group by Ref
ExtraHeaderOff	Stores the extra fields entered in the header part of the document	select u.NarrationYH, d.* from data d, u0010 u where u.ExtraId=d.Extraoff
SizeofRec	Stores the size of each record in bytes as stored in the database	
Links0	Stores the first link, depending on if its base, then reference of this value will be there in the linked document record in one of the links field	
Links1	Stores the second link, if the document has more than one base or if its linked to another document	
Links2	Stores the third link	
Links3	Stores the 4th link	
LinktoPrBatch	If a document is linked (Overheads payment) to a production batch, this field stores the batchid	
ExchRate	Stores the exchange rate of the document	
Tags4	Stores the employee (Tag5)	
Tags5	Stores the Machine (Tag6)	
Tags6	Stores the RegionId (Tag7)	
Tags7	Stores the Fixed Asset Id (Tag8)	
RMAId	Stores the Serial Number RMAId that is given for products having Serial/RMA/RFID numbers	
BinPos	Stores the position of the Bin linked to the Bins table	
BinNoEntries	Stores the number of Bins selected for the line transaction	
ReserveNo	Stores the reservation/release details for the inventory items reserved	
ReserveType	Stores the information if its Reserve or Released inventory	
NoBillwise	Stores the number of billwise adjustment against this transaction line	

Related Tables where MasterID of mr000 is linked

Table Name	Field Name	Description
Header	HeaderID	Stores the HeaderOff of data table. This table stores information header information related to each document
HEader2	HEaderId	Stores the headeroff of data table. This table stores inventory related header details of each document
Indta	SalesId	Stores the SalesOff of data table.
LoginDetails	HeaderId	Stores the login information of user entries
mr000	MasterId	Code and BookNo of data
mr001	MasterId	Linked to ProductCode in data
mr002	MasterId	Linked to tags0 in data
mr003	MasterId	linked to tags1 in data
mr004	MasterId	linked to tags2 in data
mr005	MasterId	linked to tags3 in data
mr006	MasterId	linked to tags4 in data
mr007	MasterId	linked to tags5 in data
mr008	MasterId	linked to tags6 in data
mr009	MasterId	linked to tags7 in data
mr00a	MasterId	linked thru bins table that is linked to BinPos in data
Refrn	RefId	Stores the billwiseOff of data table
rma	RMAId	Linked to RMAID in data
u0??	ExtralId	All U tables after u000a are linked to data either on ExtraOff or ExtraHeaderOff depending on the custom layout files created in documents
vtypes	VoucherType	Linked to VoucherType in data

Data Header

Field Name	Description	Sample Query
HeaderId	Stores unique Id linked to the data table	select h.*, d.* from data d, header h where d.HeaderOff=h.HeaderId
Id	Not in Use	
Version	Stores the data version	
NoEntries	Stores the number of lines entered in the document body	
TNoEntries		
Login	Stores the Login name of user who has saved the document last. To know the details of earlier editing/deleting, check headerdeleted. In Login of HeaderDeleted it will show you user1/user2 or user1-user2. If its User1/User2 then the first user1 is the user who saved it and the User2 is the user who edited it last. if its User1-User2 then User1 is the user who saved it and User2 is the user who deleted it	
Date_	Stores the date when this document was last saved. Convert the date from integer to date format using dbo.ConvertFocusDate function	

Time_	Stores the time when this document was saved. Convert using time function	select cast(Time_ as int)/(256*256)&0xff Hours, cast(Time_ as int)/(256)&0xff Minutes, cast(Time_ as Int)&0xff seconds, * from header
Flags	Stores reserve flag	
Flags2	Stores the authorization flag	Converts Pending to Authorised --- update header set Flags2=Flags2 0x200
DefCurr	stores in binary the currency of the document	
VoucherType	Stores the voucher type linked to Vtypes table	
VoucherNo	Stores the document number	
ApprovedBy0	Stores the Login Id of the user who is supposed to authorise	
ApprovedBy1	If there is multi-level authorization stores the second user who is to authorise the document	
ApprovedBy2	3rd user to authorise	

Inventory

indta

Field Name	Type	Description
SalesId	Int	This is lined to data detail on sales off. Stores unique Sales id for each row of item line
Quantity	Float	Stores the Entered/Base quantity for each item. Posted Quantity after converting to base unit when unit is used
StockValue	Float	Stores the Stock value as on posting time for issue/received document depending on all the charges added to stock from inventory screen
Rate	Float	Stores the entered Rate for the product
Gross	Float	Stores the QuantityXRate value
Qty2	Float	Stores the actual entered value when unit is used, if unit is not used or base unit is selected Qty2 and Quantity would be same
SubProcess	Float	Stores the subprocess selected in issue/receipt from production
Unit	Binary	Stores the Unit id in Binary form.
Input(N)	Float	Stores the input value for each column. N refers to number of columns defined in the inventory related screens
Output(N)	Float	Stores the output value of each column arrived at after using the formula defined in the screens.

Fixed Assets

Addvl refers to additional value added to asset, stores the details of changes in value of asset entered through Change Asset Value

Columns	Type	Description
---------	------	-------------

Vno	Varchar(21)	Stores the document number
AssetCode	Int	Stores the MasterId of Asset mr009
Date_	Int	Stores the date converted in the integer format, see date for conversion to date
BookValue	Float	Stores the original value of asset
AddValue	Float	stores the added value
commenceDate	int	stores the commencement date in integer form
Remarks	Varchar(100)	stores the remarks entered in change asset value screen

Adspl refers to asset disposal, stores the details of asset disposals.

Columns	Type	Description
Disposal No	Varchar(21)	Stores the document number
Date_	Int	Stores the Date of disposal
AssetCode	int	Stores the Masterid of the asset disposed - mr009
PurchasePrice	Float	Stores the original purchase price
AccDepre	Float	stores the depreciation till the date of disposal
SalesAccount	Int	stores the sales account masterid - mr000
QtySold	Int	stores the quantity sold
ValueSold	Float	stores the sale value

Amntn refers to asset maintenance stores the details of Asset Maintenance entered through maintenance screen

Columns	Type	Description
ContractNo	Varchar(21)	Stores the document number
AssetCode	Int	Stores the Masterid of the asset disposed - mr009
DateofCommencement	int	Stores the date of commencement of maintenance contract
DateofExpiry	int	Stores the expiry of maintenance contract
Vendor	Int	Stores the masterid of vendor account (mr000)
InitialReading	Float	Stores the reading at the time of commencement of contract
NoofUnits	Int	stores the number of units to maintain
RateofMaintenance	Float	stores the rate of maintenance per unit
ValueofMaintenance	Float	stores the value of maintenance UnitsXRate
Remarks	Varchar(200)	Stores the remarks
Doc	varchar(100)	Stores the path where the actual contract file in document or image form is located

Arepr refers to Asset Repair, stores the details of asset repairs

Columns	Type	Description
RepairNo	Varchar(21)	Stores the document number
AssetCode	Int	Stores the Masterid of the asset disposed - mr009

DateofRepair	int	Stores the date of repair
PartsAddedReplaced	Varchar(60)	stores the parts added in text form
Qty	Int	Stores the Quantity repaired
Rate	Float	Stores the rate of repair
Value	Int	Stores the value of repair
ExpiryofWarranty	Float	Stores the expiry of warranty for this repair
Remarks	Varchar(200)	Stores the remarks

Astrg abbreviation refers to Asset Register. This table stores the details of asset as entered in the Fixed Asset Master

Columns	Type	Description
MasterId	Int	Stores unique id for each asset
PurchaseDate	int	Stores the purchase date of asset in integer form
AssetType	Smallint	stores the type of asset. 0-Vehicle, 1-FixedAsset, 2-Machinery, 3-Land, 4-Building
Location	Int	Stores the MasterId of location tag.
Vendor	Int	Stores the MasterId of Vendor Account mr000
Qty	Int	Stores the quantity of the asset
ITAccruedDepreciation	int	Stores the accumulated depreciation brought forward for income tax purpose
Price	Float	Stores the original price of the asset
InsuredValue	Float	stores the insured value of the asset
ScrapValue	Float	stores the scrap value of the asset
ITDepRate	Float	stores the Rate of depreciation defined for income tax purpose
JpgPath	varchar(60)	Stores the image path
AccruedDepreciation	Float	Stores brought forward depreciation
ValueAdded	Float	Stores the value added during the year
ValueDeducted	Float	stores the value deducted through disposal during the year
DepreciationMethod	Smallint	Stores the method of depreciation, 1-Diminishing Value, 2-Diminishing value calculated halfyearly, 3-Straightline, 4-ChargeOff, 5-Usage
UnitofUsage	Float	Stores the unit of usage, Hour, Km, Production Qty, Production Runs.
RateOfDepre	Float	Stores the Rate of Depreciation
Commencement	Int	Stores the commencement date in integer form
StartingUnit	Float	Stores the Starting unit if asset is depreciation on usage
AssetAcCode	Int	Stores the MasterId of Asset Account in Account Master mr000
DepreCode	Int	Stores the MasterId of Accumulated Depreciation in Account Master mr000
DeprePLCode	Int	Stores the MasterId of Depreciation in Account Master mr000
SalesAccount	Int	Stores the MasterId of Sales Account in Account Master
Flags	Int	

Atrfr refers to asset transfer between locations entered through Transfer of Asset form

Columns	Type	Description
---------	------	-------------

TrasferNo	Varchar(21)	Stores the transfer document number
Date_	Int	stores the date in the integer form
AssetCode	int	Stores the masterid of asset mr009
FromLocation	Int	stores the masterid of location transferred from
ToLocation	Int	stores the masterid of location transferred to
Commencement Date	int	stores the original commencement date of depreciation as defined in the asset master
NewDate	Int	stores the new date of commencement from this location
Value	Float	Stores the value at the time of transfer

ausag refers to Asset Usage and stores information as entered in the Asset Usage form		
Columns	Type	Description
UsageNo	Varchar(21)	Stores the transfer document number
Date_	Int	stores the date in the integer form
AssetCode	int	Stores the masterid of asset mr009
Unit	Float	Stores the unit of measurement for usage
LastMeterReading	Float	stores the last meter reading information
NoofNewUnits	Float	stores the number of new units used
NewMeterReading	Float	stores the new meter reading value
PRoductionBatch No	varchar(50)	stores the production batch if usage is of production based
ProductManufactured	int	stores the MasterId of the product mr001
QtyManufactured	Float	stores the quantity manufactured

u0009 stores the additional custom field information if any created in Fixed Asset Master		
Columns	Type	Description
ExtraID	Int	Stores the ExtraId linked to Eoff in mr009 table

Point of Sales (POS)

PosHeader table is used to store header information of bill.		
Columns	Type	Description
BillNo	int	Stores bill number without prefix.
HeaderID	int	Stores bill number without prefix.
BillDate	varchar(10)	Stores bill date.
SalesCode	int	Stores Sales A/c.
CustCode	int	Stores customer code.
CashierCode	varchar(20)	Stores cashier name.

CounterCode	int	Stores counter id.
NetAmount	numeric(12, 2)	Stores net amount.
GrossAmount	numeric(12, 2)	Stores gorss amount.
InputQty	bit	Stores info about quantity is editable during billing.
InputRate	bit	Stores info about rate is editable during billing.
InputCust	bit	Not in use.
Tax	numeric(10, 2)	Total tax on bill.
Discount	numeric(10, 2)	Total discount on bill.
Spl_Disc	numeric(10, 2)	Overall discount on net amount.
Currency1	smallint	Stores first currency id used in cash payment.
Currency2	smallint	Stores second currency id used in cash payment.
Currency3	smallint	Stores third currency id used in cash payment.
Currency4	smallint	Stores fourth currency id used in cash payment.
ExchangeRate1	numeric(10, 5)	Stores exchange rate of currency 1.
ExchangeRate2	numeric(10, 5)	Stores exchange rate of currency 2.
ExchangeRate3	numeric(10, 5)	Stores exchange rate of currency 3.
ExchangeRate4	numeric(10, 5)	Stores exchange rate of currency 4.
CashAmount1	numeric(12, 2)	Stores cash amount paid by currency 1.
CashAmount2	numeric(12, 2)	Stores cash amount paid by currency 2.
CashAmount3	numeric(12, 2)	Stores cash amount paid by currency 3.
CashAmount4	numeric(12, 2)	Stores cash amount paid by currency 4.
CreditAmount	numeric(12, 2)	Stores cheque amount used in bill payment.
CreditCardAmount	numeric(12, 2)	Stores credit card amount used in bill payment if multiple card not supported. Otherwise 0.
Card_Name	varchar(50)	Stores card name used in bill payment if multiple card not supported. Otherwise 0..
Card_Type	varchar(50)	Stores card type used in bill payment if multiple card not supported. Otherwise 0.
Card_Number	varchar(50)	Stores card number used in bill payment if multiple card not supported. Otherwise 0.
Expiry_Date	varchar(10)	Stores card expiry date used in bill payment if multiple card not supported. Otherwise 0.
ApprovalCode	varchar(50)	Not in use.
TranTime	varchar(50)	Stores current time.
TranMode	smallint	Stores information about bill suspended.
TranMark	smallint	Stores information about bill suspend screen.
PostStatus	smallint	Stores information about data posting.
CustCode2	int	Stores customer code.
IsCustTag	bit	Stores information about customer is selected from tag.
BillAmount	numeric(12, 2)	Stores bill amount which customer has to pay.
flags	int	Stores information about home delivery.
voucher_name	varchar(50)	Not in use.
voucher_amount	numeric(12, 2)	Stores amount paid using voucher point.
voucher_number	varchar(50)	Not in use.
cheque_name	varchar(50)	Stores bank name used in cheque payment.
cheque_amount	numeric(12, 2)	Stores amount used in cheque payment.

cheque_date	datetime	Stores cheque date used in cheque payment.
cheque_number	varchar(50)	Stores cheque number used in cheque payment.
cashvoucher_amount	numeric(12, 2)	Stores total coupon amount used in bill payment.
SalesmanID	int	Stores salesman id.
ShiftID	smallint	Stores shift id.
Remarks	varchar(30)	Stores remark given in entry screen.
dSchemeSavingAmount	numeric(10, 2)	Stores total scheme saving amount.

PosCreditCardPayment table is used to store card payment information if multiple card payment option is enabled.

Field Name	Type	Description
HeaderID	int	Stores headerid of bill. Related to PosHeader table.
CreditCardNumber	varchar(30)	Stores card number used in bill payment.
CreditCardName	varchar(30)	Stores card name used in bill payment.
CreditCardType	varchar(30)	Stores card type used in bill payment.
ExpiryDate	varchar(10)	Stores expiry date of card used in payment.
Amount	numeric(12, 2)	Stores card amount used to pay bill.

Pos_Header_CashVoucher table is used to store coupon information used in bill payment.

Field Name	Type	Description
HeaderID	int	Stores HeaderID of bill. Related to PosHeader table.
voucherID	int	Stores voucherid(CouponId) used in bill payment.
denomination	numeric(12, 3)	Stores each voucher amount.
qty	int	Stores quantity of denomination.

PosHdr_HomeDelivery table is used to store home delivery data if payment home delivery option is used in pos billing.

Field Name	Type	Description
HeaderID	int	Stores HeaderID of bill. Related to PosHeader table.
Address	varchar(512)	Stores home delivery address.
DlvDate	datetime	Stores delivery date.

PointsAchieve table is used to store total accumulated points in bill.

Field Name	Type	Description
BillNo	int	Stores bill number. Related to PosHeader table.
BillDate	varchar(10)	Stores bill number.
CustCode	int	Stores customer id.
TotalPoints	numeric(10, 2)	Stores total points accumulated in bill.
IsCustTag	bit	Is customer is selected from tag.

Line table is used to store detail information of bill.

Field Name	Type	Description
HeaderID	int	Stores HeaderId of bill. Related to PosHeader table.
ProdCode	int	Stores product id. Related to mr001
ProdRate	numeric(10, 2)	Stores rate of product.
ProdQty	decimal(10, 4)	Stores quantity of product.
A	numeric(10, 2)	Stores discount of product.
B	numeric(10, 2)	Stores tax of product.
C	numeric(10, 2)	Not in use.
D	numeric(10, 2)	Not in use.
E	numeric(10, 2)	Not in use.
F	numeric(10, 2)	Not in use.
RefundFlag	bit	Stores information whether this item returned form entry screen.
Remarks	varchar(60)	Stores discount remark.
Tag2ID	int	Stores size.
SalesAc	int	Stores Sales A/c.
RateType	smallint	Stores wholesale or retail rate information if cut off retail is set in pos preferences.
DiscountInPercentage	bit	Stores if discount is given in percentage.
Spl_Disc	decimal(10, 3)	Stores special discount given to product. Used only in MICROTEx client.
iBodyID	bigint	Auto generated number.
RateRemark	varchar(60)	Stores rate remark.

Pos_lineRMA table is used to store RMA number of sold product.

iBodyID	bigint	Stores bodyId. Related to Line table.
SerialNumber	varchar(30)	Stores serial number.

ItemReturns table is used to store returned product information.

Field Name	Type	Description
DocID	int	Stores document id.
ProdCode	int	Stores productid of returned product. Related from mr001.
ProdRate	float(53)	Stores product rate of returned product.
ProdQty	float(53)	Stores product quantity of return product.
A	float(53)	Stores discount of returned product.
B	float(53)	Stores tax of returned product.
C	float(53)	Not in use.
D	float(53)	Not in use.
E	float(53)	Not in use.
F	float	Not in use.

BillDate	int	Stores return date.
CashierCode	varchar(20)	Stores cashier name.
CounterCode	int	Stores counter id.
Flag	int	Stores information about data posting.
ReturnMode	int	Stores information about exchange goods for coupon, cash, credit.
billNo	varchar	Stores bill number with prefix if item is returned from return screen.
CustCode2	int	Stores customer id.
ShiftID	int	Stores shift id.
iID	int	Auto generated number.

PosSchemeApplied table is used to store applied schemes on bill.

Field Name	Type	Description
iSchemeID	int	Stores applied scheme id. Related to Pos_SchemeHeader table.
iHeaderID	int	Stores headerid of bill on which scheme is applied. Related to PosHeader table.

PayFrmPoint table is used to store coupons such as sodexho.

Field Name	Type	Description
voucherID	int	Stores voucher id(CouponId).
voucherName	varchar(50)	Stores coupon name.

Cash_in_Out table is used to store information of voucher point is used during bill payment.

Field Name	Type	Description
BillNo	int	Stores bill number. Related to PosHeader table.
RefDoc	int	Stores reference doc id.
AmountUsed	numeric(10, 2)	Stores amount used in payment.

VoucherConvert table is used to store cash in and cash out data.

Field Name	Type	Description
VoucherNo	int	Stores voucher number.
VchrType	int	Stores voucher type information means this is cash receipt or cash payment information.
AccountId	int	Stores cash account id. Related to mr000 table.
CashierCode	varchar(20)	Stores cashier name.
CounterId	int	stores counter id.
Amount	numeric(12, 2)	Stores cash in or out amount.
Remarks	varchar(255)	Stores remark.
TranDate	varchar(20)	Stores current date.
IsCustTag	bit	Stores information about customer is selected from tag.
Flag	int	Stores 0.

ExpenseAc	int	Stores expense A/c id. Related to mr000 table.
-----------	-----	--

RedeemPoints table stores information related to loyalty points redemption		
Field Name	Type	Description
DocID	int	Stores document id.
DocDate	varchar(10)	Stores date.
CustCode	int	Stores customer id.
CoupenNo	varchar(10)	Stores coupon number.
VoucherValue	numeric(10, 2)	Stores inputted cash in case of advance against sales. Stores inputted points * discount per point in case of redeem points. Stores return amount in case of goods return.
RefType	int	Stores id of advance against sales, redeem points, or goods return.
RefDoc	int	Stores reference doc id.
IsCustTag	bit	Stores information, if customer is tagged to the document or not.
ExpDate	datetime	Stores date in case of advance against sales and redeem points.
flags	int	Stores other information in case of advance against sales, or redeem points.

Pos_Cash Authentication1 table is used to store shift closing information, if update inventory online option is on in pos preference.		
ID	int	Auto generated number.
LoginID	int	Stores user id.
ShiftID	int	Stores shift id.
CounterCode	int	Stores counter id.
iLoginDate	int	Stores login date.
CashIn	decimal(10, 2)	Stores cash in amount during shift opening.
iLogoutDate	int	Stores logout date.
CashOut	decimal(10, 2)	Stores cash out amount during shift closing.
dCreditCardAmount	decimal(10, 2)	Stores credit card amount during shift closing.
dCouponAmount	decimal(10, 2)	Stores coupon amount during shift closing.

Pos_SchemeHeaderInfo table is used to store information about all schemes.		
Field Name	Type	Description
iSchemeType	int	Stores scheme type to determine whether this is discount or free gift.
iSchemeID	int	Auto generated number.
SchemeName	varchar(25)	Stores scheme name.
iStartingDate	int	Stores scheme starting date.
iStartingTime	int	Stores scheme starting time.
iEndingDate	int	Stores scheme ending date.
iEndingTime	int	Stores scheme ending time.
dAmount	numeric(12, 2)	Stores amount on which applied scheme will depend.

bExclusive	bit	Stores information Is this scheme can be combined other schemes.
iLocation	int	Stores location where this scheme will be applicable.

Pos_SchemeDependency table is used to store information if scheme condition is depend on products.

iSchemeID	int	Stores scheme id. Related to Pos_SchemeHeader table.
iProductID	int	Stores product id on which applied scheme will depend. Related to mr001 table.
dMin	decimal(5, 2)	Stores minimum quantity of product.
dMax	decimal(5, 2)	Stores maximum quantity of product.

Pos_SchemeDetail table is used to store combined scheme information.

iSchemeID	int	Stores scheme id which is combined other schemes. Related to Pos_SchemeHeader table.
iCombinedScheme	int	Stores scheme id of combined scheme. Related to Pos_SchemeHeader table.

PosSchemeDetailFreeGift stores information about free gift schemes

iSchemeID	int	Stores scheme id. Related to Pos_SchemeHeader table.
GroupName	varchar(15)	Stores group name which is used to categorised free gift and gives facility to select free gift amount if any.
iProductID	int	Stores free gift product. Related to mr001 table.
dQuantity	decimal(7, 3)	Stores free gift quantity.

Pos_CardType table is used to store available card type such as MASTER, VISA etc. Used if multiple card payment option is enabled.

CardType	varchar(10)	Stores card type such as MASTER, VISA etc.
----------	-------------	--

UserLogin table is used to store user information each time go into pos module.

UserName	varchar(20)	Stores login user name.
LoginDate	varchar(10)	Stores login date.
LoginTime	varchar(10)	Stores login time.
LogoutTime	varchar(10)	Stores logout time.

PrintSetup table is used to store print format of bill printing.

PrintMode	int	Stores information about is print item name in next line.
HeaderInfo	varchar(2000)	Stores information print in header section of bill.
ProdCodePos	int	Stores information about product code position in body section.
ProdCodeWidth	int	Stores product code width in bill.
ProdNamePos	int	Stores information about product name position in body

		section.
ProdNameWidth	int	Stores product name width in bill.
ProdQtyPos	int	Stores information about product quantity position in body section.
ProdQtyWidth	int	Stores product quantity width in bill.
ProdRatePos	int	Stores information about product rate position in body section.
ProdRateWidth	int	Stores product rate width in bill.
ProdDiscPos	int	Stores information about discount position in body section.
ProdDiscWidth	int	Stores discount width in bill.
ProdTaxPos	int	Stores information about tax position in body section.
ProdTaxWidth	int	Stores tax width in bill.
ProdNetPos	int	Stores information about net position in body section.
ProdNetWidth	int	Stores net width in bill.
ProdGrossPos	int	Stores information about gross position in body section.
ProdGrossWidth	int	Stores gross width in bill.
FooterInfo	varchar(2000)	Stores information print in footer section of bill.
E0	int	Stores information about extra field position of product in body section.
EW0	int	Stores extra product width in bill.
E1	int	Stores information about extra field position of product in body section.
EW1	int	Stores extra product width in bill.
E2	int	Stores information about extra field position of product in body section.
EW2	int	Stores extra product width in bill.
E3	int	Stores information about extra field position of product in body section.
EW3	int	Stores extra product width in bill.
E4	int	Stores information about extra field position of product in body section.
EW4	int	Stores extra product width in bill.
E5	int	Stores information about extra field position of product in body section.
EW5	int	Stores extra product width in bill.
E6	int	Stores information about extra field position of product in body section.
EW6	int	Stores extra product width in bill.
E7	int	Stores information about extra field position of product in body section.
EW7	int	Stores extra product width in bill.
E8	int	Stores information about extra field position of product in body section.
EW8	int	Stores extra product width in bill.
E9	int	Stores information about extra field position of product in body section.
EW9	int	Stores extra product width in bill.
ProdDiscPosPer	int	Stores information about product discount in percent position in body section.

ProdDiscPerWidth	int	Stores product discount in percent width in bill.
ProdTaxPosPer	int	Stores information about product tax in percent position in body section.
ProdTaxPerWidth	int	Stores product tax in percent width in bill.
ProductAlias	int	Stores information about product alias position in body section.
ProductAliasWidth	int	Stores product alias width in bill.
SnoPos	int	Stores information about serial number position in body section.
SnoWidth	int	Stores serial number width in bill.
TAG2Size	int	Stores information about size position in body section.
TAG2SizeWidth	int	Stores size width in bill.
MRPPos	int	Stores information about MRP position in body section.
MRPWidth	int	Stores MRP width in bill.
SavingPos	int	Stores information about saving amount on item position in body section.
SavingWidth	int	Stores saving amount on item width in bill.
RMAPos	int	Stores information about RMA position in body section.
RMAWidth	int	Stores RMA width in bill.
RemarkPOS	int	Stores information about Remark position in body section.
RemarkWidth	int	Stores Remark width in bill.
HeaderImage	image	Not in use.
iLocation	int	Stores location of print format.
iDetailLinesPerPage	int	Stores number of body lines per page.

PrintSetupGR table is used to store print format of goods return bill printing.		
PrintMode	int	Stores information about is print item name in next line.
HeaderInfo	varchar(2000)	Stores information print in header section of bill.
ProdCodePos	int	Stores information about product code position in body section.
ProdCodeWidth	int	Stores product code width in bill.
ProdNamePos	int	Stores information about product name position in body section.
ProdNameWidth	int	Stores product name width in bill.
ProdQtyPos	int	Stores information about product quantity position in body section.
ProdQtyWidth	int	Stores product quantity width in bill.
ProdRatePos	int	Stores information about product rate position in body section.
ProdRateWidth	int	Stores product rate width in bill.
ProdDiscPos	int	Stores information about discount position in body section.
ProdDiscWidth	int	Stores discount width in bill.
ProdTaxPos	int	Stores information about tax position in body section.
ProdTaxWidth	int	Stores tax width in bill.
ProdNetPos	int	Stores information about net position in body section.
ProdNetWidth	int	Stores net width in bill.
ProdGrossPos	int	Stores information about gross position in body section.

ProdGrossWidth	int	Stores gross width in bill.
FooterInfo	varchar(2000)	Stores information print in footer section of bill.
E0	int	Stores information about extra field position of product in body section.
EW0	int	Stores extra product width in bill.
E1	int	Stores information about extra field position of product in body section.
EW1	int	Stores extra product width in bill.
E2	int	Stores information about extra field position of product in body section.
EW2	int	Stores extra product width in bill.
E3	int	Stores information about extra field position of product in body section.
EW3	int	Stores extra product width in bill.
E4	int	Stores information about extra field position of product in body section.
EW4	int	Stores extra product width in bill.
E5	int	Stores information about extra field position of product in body section.
EW5	int	Stores extra product width in bill.
E6	int	Stores information about extra field position of product in body section.
EW6	int	Stores extra product width in bill.
E7	int	Stores information about extra field position of product in body section.
EW7	int	Stores extra product width in bill.
E8	int	Stores information about extra field position of product in body section.
EW8	int	Stores extra product width in bill.
E9	int	Stores information about extra field position of product in body section.
EW9	int	Stores extra product width in bill.
ProdDiscPosPer	int	Stores information about product discount in percent position in body section.
ProdDiscPerWidth	int	Stores product discount in percent width in bill.
ProdTaxPosPer	int	Stores information about product tax in percent position in body section.
ProdTaxPerWidth	int	Stores product tax in percent width in bill.
ProductAlias	int	Stores information about product alias position in body section.
ProductAliasWidth	int	Stores product alias width in bill.
SnoPos	int	Stores information about serial number position in body section.
SnoWidth	int	Stores serial number width in bill.
TAG2Size	int	Stores information about size position in body section.
TAG2SizeWidth	int	Stores size width in bill.
MRPPos	int	Stores information about MRP position in body section.
MRPWidth	int	Stores MRP width in bill.
SavingPos	int	Stores information about saving amount on item position in

		body section.
SavingWidth	int	Stores saving amount on item width in bill.
RMAPos	int	Stores information about RMA position in body section.
RMAWidth	int	Stores RMA width in bill.
RemarkPOS	int	Stores information about Remark position in body section.
RemarkWidth	int	Stores Remark width in bill.
iLocation	int	Stores location of print format.
iDetailLinesPerPage	int	Stores number of body lines per page.

PosPref table is used to store global settings on POS.		
DriverName	int	Stores information about which driver is used for POS.
DrawerPort	int	Stores information about cash drawer port.
PrinterPort	int	Stores information about printer port.
LineDispPort	int	Stores information about line display port.
PostingMode	int	Stores information how data will be posted.
SalesAccountID	int	Stores Sales A/c. Related to mr000.
CashAccountID	int	Stores Cash A/c. Related to mr000.
DebitAccountID	int	Stores Debit A/c. Related to mr000.
CreditCardAccountID	int	Stores Card A/c. Related to mr000.
DrAct_CrSal	int	Stores Cheque A/c. Related to mr000.
CoupenID	int	Stores Coupon A/c. Related to mr000.
SuspenseID	int	Stores Suspense A/c. Related to mr000.
CounterID	int	Stores which tag counter (location) is related.
SplDiscID	int	Stores Special Discount A/c. Related to mr000.
TaxID	int	Stores Tax A/c. Related to mr000.
DiscountID	int	Stores Discount A/c. Related to mr000.
TrackCustPoint	bit	Stores information Is Point tracking system is enabled.
CustTagID	int	Stores which tag customer is related.
PurchasePoints	numeric	Stores how much purchase get one point.
DiscountPoint	numeric	Stores how much discount will get per point.
FieldName	varchar	Stores Cashier Name,Credit Card No,Credit Card Name,Credit Card Type,Expiry Date. Not in use.
Identifier	varchar	Stores value of fields. Not in use.
IsCustTag	bit	Stores Is customer is from tag.
AmtRound	smallint	Stores amount round off information.
prefix_seq	varchar	Stores prefix of bill number.
ReturnApproval	smallint	Stores information about is manager authentication will be asked when item is returned.
UpdateInventoryOnline	smallint	Stores information Is bill will be posted as soon as it is saved.
DevicePort	smallint	Stores weight machine port.
VATInclusive	smallint	Stores is VAT is inclusive.
RoundOffAccount	int	Stores Round Off A/c. Related to mr000.
CutOffRetail	smallint	Stores whether wholesale or retail rate will be applied.
Code2	smallint	Use to get new bill sequence number. Presently not in use.
ShowMRP	bit	Stores information Is MRP will be showed during billing.

PickEODAmount	bit	Stores information about automatically pick EOD amount.
ShowSalesman	bit	Stores information is salesman selection will be enabled during billing.
CustomerSelectionScreen	bit	Stores information about customer selection screen(Entry or Payment).
NewLoyaltyPointScheme	bit	Not in use.
LoyaltyDiscountOnSameBill	bit	Not in use.
bCouponPayment	bit	Stores information about coupon payment enable.
bCreditCardPayment	bit	Stores information about card payment enable.
bChequePayment	bit	Stores information about cheque payment enable.
iMachineType	int	Stores information about weight machine type.
iGainAccountID	int	Not in use.
bAllowSalesReturnEditing	bit	Not in use.
bShiftWisePosting	bit	Not in use.
iSalesReturnAc	int	Stores Sales Return A/c. Related to mr000.

Human Resources Management (HRM)

Payroll Preferences

Payroll preferences table			
TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxPpref			
	CatId	int	Category id
	Flags	int	in this column, 1'st bit indicates =>"Hide page totals in Reports", 2'nd bit indicates =>"sitewise summary posting" , 3'rd bit indicates=>"Round net amount to nearest value", 4'th bit indicates=>"Add HOT to Vacation", 5'th to 31 bit will hold "EndOfService Loans", last bit indicates=>"Hold employee salary"
	NewFlags	int	in this column, 1'st bit indicates=>"include authorization in vacation leave application", 2'nd bit indicates=>"HOS Loan Approval is required" , 3'rd bit indicates=>"Attendance by employee group", 4'th bit indicates=>"Donot maintain usage log", remaining bits are not used....
	iVacLvType	int	sequence id of Leave which is considered as Vacation leave
	szEBFormula	varchar	earned basic formula
CxxxLvPref			Leaves Preferences
	CatID	int	Category id
	Flags	int	in this column, 1'st bit indicates =>HOSApprovalNotRequired, 2'nd bit indicates GMApprovalNotRequired , 3'rd to 8'th bit indicates DivideByDaysForEncash , 9'th to 17'th bit will hold

			MinimumNumberOfDaysForGMAApproval, remaining bit's are not used
	Formula	nvarchar	Leave Encashment basic Formula
	Decoded	nvarchar	internal formula for Leave encashment basic
CxxxRights			this table will store Rights which is assigned to Particular focus user to access departments & category
	UserID	int	focus user id
	CatId	int	this can store either category id or department id based on value of Type
	iAccess	int	it holds either 0 or 1, it indicates whether corresponding user has rights to access that particular catid/dept information
	Type	int	this will take 2 values I,e... 1,0. if Type=1 then CatId column is holding department id. If Type=0 then CatId Column is holding categoryId

Employee Master

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxEmpDetails			
	Code	INT	unique code
	Empld	int	employee sequence id from focus
	BasicSal	float	basic salary
	Basic1	float	offshore basic salary(applicable for MiddleEast)
	Category	int	category from payroll preferences
	Currency	int	currencyid in focus CurrencyRecord
	Curren1	int	Currencyid for vacation basic
	DateofJoin	DATE	joining date
	DateofRev	DATE	maintains dates of salary revisions
	ApplicableDate	DATE	if employee has any revision and applicable date is less than dateofrev employee is eligible for arrears
	Dept	int	dept id
	AppraisalDate	date	Next appraisal date used for expiry based report
	EmployeeStatus	int	maintains employee status(0-OnDuty,1-OnVacation,2-Resigned,3-Suspended,4-Terminated, 5=>deceased, 6=>absconded)
	HalfDay	int	weekly halfday (sequence of week days .1-Sunday,2-Monday etc...)
	HoliDay1	int	weekly off (sequence of week days .1-Sunday,2-Monday etc...)
	HoliDay2	int	weekly off (sequence of week days 1-Sunday,2-Monday etc...)
	LastVacDate	date	Last vacation date
	Location	int	location where employee is working
	PaybleAccount	int	account to post
	SalAccount	int	salary account
	Shift	int	not used

	PhotoPath	varchar	photopath
	ResignationDate	int	If employee resigns, then this will store resignation date
	ResignReason	varchar	not used
	Sector	int	sector id loads from sector master. Used for air ticket
	TravelClass	int	travel class from sector master.1- "Economy",2-"Business",3-"First-class"
	State	int	state id loads from PF preferences
	VacationDueOn	date	next vacation date maintained for expiry based report
	VacLvs	float	no of leaves eligible for vacation
	VacPerDays	float	vacation per days (ex per 360 working days 30 leaves)
	VacLvs1	float	no of leaves eligible for vacation (offshore)
	VacPerDays1	float	vacation per days (ex per 180 offshore working days 30 leaves)
	VacLvsBf	float	brought forwarded vacation leaves
	VacLvsOffBf	float	brought forwarded vacation leaves (offshore)
	VacFlags	smallint	it stores information of AddWorkingWeeklyOffToAnnualLeave, AddWorkingPublicHolidaysToAnnualLeave, AddOffShoreWorkingWeeklyOffToAnnualLeave and AddOffShoreWorkingPublicHolidaysToAnnualLeave
	TDaysWorkedBfLastVac	float	to calculate no of days eligible for vacations
	TDaysWorkedBfLastVacOff	float	to calculate no of days eligible for vacations
	TDaysWorkedBf	float	to calculate gratuity
	TDaysWorkedOFFBf	float	to calculate gratuity
	AirTicket	float	to calculate air tickets eligibility
	AirPerDays	float	criteria to calculate air tickets (ex per360 days 1 air ticket)
	AirTicketOff	float	to calculate Offshore air tickets eligibility
	AirOffPerDays	float	offshore criteria to calculate air tickets (ex per360 days 1 air ticket)
	AirTicketBf	float	tickets brought forwarded
	AirTicketOffBf	float	offshore air tickets brought forwarded
	BioDataID	int	Biodata Format ID
	BioDataRef	varchar	biodata reference id ,If employee joins from recruitment
	BankAcc	int	Bank account Id from account master
	VacId	int	Vacation Category Id
	LastLOP	float	LOP on Last Vacation
	TotalLOP	float	Total LOP b/f
	TotalLvsUtilised	float	Total Leaves Used
	TotalHW	float	Total Holidays Worked b/f
	NoticeDate	int	Notice Date
	NoticePeriod	float	notice Period
	ProbationPeriod	float	Probation period in months
	DOB	int	date of birth
	DOM	int	Date of marriage
	DrivingLicenceExpDt	int	Driving license expiry date
	PassportExpDt	int	passport expiry date

	VisaStDt	int	Visa starting date
	VisaEndDt	int	Visa expiry date
	Gender	int	0=>Male, 1=>female
	MaritalStatus	int	0=>married, 1=>unmarried, 2=>Divorced, 3=>widow
	VisaType	int	0=>visiting, 1=>employment, 2=>Government, 3=>Contract, 4=>TempWorkPermit
	DrivingLicenceNo	nvarchar	Driving license number
	CurrentAddr	nvarchar	Current address
	CurrentAddrCity	nvarchar	Current address City
	CurrentAddrPin	nvarchar	Current address Pin
	PhoneNo	nvarchar	phone number
	Nationality	nvarchar	Nationality
	VisaNo	nvarchar	visa number
	PassportNo	nvarchar	Passport number
	PermanentAddr	nvarchar	Permanent address
	PermanentAddrCity	nvarchar	Permanent address City
	PermanentAddrCountry	nvarchar	Permanent address Country
	PermanentAddrPin	nvarchar	Permanent address Pin
	EMailID	nvarchar	Email id
	BankBranch	nvarchar	Branch name
	AccNo	nvarchar	Bank account number
	iLastAirTktDate	int	Last air ticket date
	iSpouseType	int	1=>if spouse is working in same company, 0=>spouse is working in different company
	SpouseEmpId	int	if spouse is working in same company, this will hold ID
	szSpouseName	nvarchar	if spouse is working in different company, this will store Name of the spouse
	AirTktEntitlement	float	Air ticket entitlement period
	CivilId	varchar	Civil ID (used by CBK)
	CivilDate	int	Civil Date (used by CBK)
	iLastEditDt	bigint	this will hold previously edited date and time
	szLastUser	nvarchar	this will hold user name who edited this employee record
	iExtraId	int	Extra field id
	iContrExpDt	int	Contract expiry date
	iContrNoticePeriod	int	when contract notice period is reached, then alerts will be sent to higher officials
	UserName	nvarchar	Login name. (this column is not used It will be used in FOW)
	Pwd	nvarchar	Login Password. (this column is not used It will be used in FOW)
	szMobileNum	varchar	mobile number
	Insured	tinyint	1=>insurance provided
	szDesignation	varchar	designation
	iPPIssueDt	int	passport issue date
	szPPIssPlace	varchar	Passport issue place
	szVisalssPlace	varchar	Visa Issue place
	szVisaComp	varchar	Visa company
	iDrvLiceType	smallint	Driving license type
	szDrvLicePlace	varchar	Driving license issue place
CxxxRelativeinfo			employee relatives information
	ild	int	unique code

	Empld	int	Employee id
	RevDate	int	Date of revision. This will refer to DateOfRev column in CxxxEmpDetails table
	iRelEmpld	int	ID of employee relative
CxxxRelativeInfo			Relative Information
	ild	int	unique code
	Empld	int	Employee ID
	RevDate	int	Date of revision. This will refer to DateOfRev column in CxxxEmpDetails table
	iRefEmpld	int	not used
	RefPath	nvarchar	file Path
	szRefEmpName	nvarchar	Name
CxxxDependantInfo			details of Employee dependants
	Empld	int	Employee Id
	RevDate	int	Date of revision. This will refer to DateOfRev column in CxxxEmpDetails table
	DependantName	nvarchar	Dependant name
	Relation	nvarchar	Relationship
	DOB	int	date of birth
	PassportNo	nvarchar	Passport number
	PassportExpDt	int	passport expiry date
	VisaNo	nvarchar	Visa Number
	VisaExpDt	int	Visa expiry date
	iMedical	int	1=>if dependant has medical insurance
	iAirTicket	int	1=>if dependant has Airticket
CxxxEarmat			earnings information
	EmpID	int	Employee id
	Code	int	code related to Code in EmpDetails table
	DateAppl	date	Date of Applicable
	DateofRev	date	Date of revision. This will refer to DateOfRev column in CxxxEmpDetails table
	E0	float	E series columns are created dynamically, depending no of earnings created in payroll preferences
	E1	float	the values which is entered in EmployeeMaster->earnings tab, will be stored in these columns
		
CxxxDedmat			Deduction information
	EmpID	int	Employee Id
	Code	int	code related to Code in EmpDetails table
	DateAppl	date	Date of Applicable
	DateofRev	date	Date of revision. This will refer to DateOfRev column in CxxxEmpDetails table
	D0	float	D series columns are created dynamically, depending no of deductions created in payroll preferences
	D1	float	the values which is entered in EmployeeMaster->Deductions tab, will be stored in this table
		

CxxxLoansMat			Loans information
	EmpID	int	Employee ID
	Code	int	code related to Code in EmpDetails table
	DateAppl	date	Related to DateofAppl in EmpDetails
	DateofRev	date	Related to DateofRev in EmpDetails
	LN0	int	LN series columns are created dynamically, depending no of Loans created in payroll preferences
	LN1	int	this will hold id of Account to which loan amount should be posted
		
CxxxLvsMat			Leaves information
	EmpID	int	Employee id
	Code	int	code related to Code in EmpDetails table
	DateAppl	date	Related to DateofAppl in EmpDetails
	DateofRev	date	Related to DateofRev in EmpDetails
	LV0	float	LV,LVEN series columns are created dynamically, depending no of Loans created in payroll preferences,LV columnsmaintains leavesBf and LVEN maintains Leaves Entitled
	LVEN0	float	
		

Monthly Payroll

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxMonthlyPayroll			
	MonthlyId	int	primary key
	PayMonthYear	datetime	month for which payroll is processed
	WorkingDays	float	no of days for that month
	Category	int	category id
	Dept	int	department id. This will be helpful when attendance is entered department wise
CxxxMonthlyPayChild			
	rIndex	int	ignore , has no meaning
	Remarks	nvarchar	brief description if any
	MonthlyId	int	has relation with MonthlyId in CxxxMonthlyPayroll
	EmpId	int	Employee id
	Location	int	has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. If employee worked in more than one site on a particular month, then this column will maintain location where employee has worked
	OtHours	float	over time hours
	HolidayOT	float	Holiday overtime hours
	NoLate1	smallint	number of late
	NoLate2	smallint	number of late
	DaysWorked	float	Number of days worked

	d		
	Dept	int	department id
	Phase	int	this has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. This column will store tag-7 id which will be useful in Region wise posting
	MPayCode	int	primary key has one to one relation with earusg,dedusg,paytable,loansusg,lvsusg
	OffShore	float	Offshore Hours Worked
	OffOT	float	Offshore Hours OT Worked
	OffHOT	float	Offshore HOT Hours Worked
	LSeq	int	Location Sequence Number
	OTHours2	float	Second OT hrs
	Designation	nvarchar	Stores the designation information
	Paid	int	0=>for unpaid employee. 1=>paid employee
	ML0	float	Dynamically created monthly inputs columns. The value which is given for Monthly Input will be stored in this column
	ML0REM	nvarchar	Dynamically created monthly inputs remarks. Monthly input remarks will be stored in this column
	ML1	float	
	ML2REM	nvarchar	
		
CxxxPayTable			
	MonthlyID	int	has relation with MonthlyId in CxxxMonthlyPayroll
	Empld	int	Employee id
	Basic	float	Earned basic
	Ot	float	Over time amount. The formula for Overtime will be evaluated and stored in this column
	Hot	float	Holiday Over time amount. The formula for Holiday Overtime will be evaluated and stored in this column
	Net	float	Net amount
	DaysCut	tinyint	No. of Days Deducted
	Dept	int	Department id
	Location	int	has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. If employee worked in more than one site on a particular month, then this column will maintain location where employee has worked
	Phase	int	this has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. This column will store tag-7 id which will be useful in Region wise posting
	MPayCode	int	primary key has one to one relation with earusg,dedusg,paytable,loansusg,lvsusg
	BankAcc	int	Bank Id from Account Master
	Designation	nvarchar	Designation Information
CxxxEarUsg			
	MPayCode	int	primary key has one to one relation with earusg,dedusg,paytable,loansusg,lvsusg
	EmpID	int	employee id
	DateofRev	datetime	month for which payroll is processed

	Location	int	has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. If employee worked in more than one site on a particular month, then this column will maintain location where employee has worked
	MonthlyID	int	has relation with MonthlyId in CxxxMonthlyPayroll
	Phase	int	this has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. This column will store tag-7 id which will be useful in Region wise posting
	Dept	int	department id
	E0	float	the formula for Earnings will be evaluated and stored in this column
	E1	float	
	E2	float	
CxxxDedUsg			
	MPayCode	int	primary key has one to one relation with earusg,dedusg,paytable,loansusg,lvsusg
	EmpID	int	Employee id
	DateofRev	datetime	month for which payroll is processed
	Location	int	has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. If employee worked in more than one site on a particular month, then this column will maintain location where employee has worked
	MonthlyID	int	has relation with MonthlyId in CxxxMonthlyPayroll
	dept	int	department id
	Phase	int	this has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. This column will store tag-7 id which will be useful in Region wise posting
	D0	float	dynamically created columns for storing deductions
	cfD0	float	dynamically created columns for storing deductions
	D1	float	Formula for deductions will be evaluated and stored in this columns
	cfD1	float	
	D2	float	
	cfD2	float	
		
CxxxLoansUsg			
	MPayCode	int	primary key has one to one relation with earusg,dedusg,paytable,loansusg,lvsusg
	EmpID	int	Employee id
	DateofRev	datetime	month for which payroll is processed
	Location	int	has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. If employee worked in more than one site on a particular month, then this column will maintain location where employee has worked
	MonthlyID	int	has relation with MonthlyId in CxxxMonthlyPayroll
	VacationCode	int	if loans paid in vacation it will has some value
	UFLAG	smallint	

	Phase	int	this has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. This column will store tag-7 id which will be useful in Region wise posting
	Dept	int	department id
	LN0	float	dynamically created columns for storing Loans
	LN1	float	Loan paid
		
CxxxLvsUsg			
	MPayCode	int	has relation with Mpaycode in CxxxMonthlyPayChild, CxxxEarusg, CxxxDedUsg, CxxxLoansUsg, CxxxLvsUsg
	EmpID	int	Employee id
	DateofRev	datetime	month for which payroll is processed
	Location	int	has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. If employee worked in more than one site on a particular month, then this column will maintain location where employee has worked
	MonthlyID	int	has relation with MonthlyId in CxxxMonthlyPayroll
	Phase	int	this has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. This column will store tag-7 id which will be useful in Region wise posting
	VacationCode	int	if leaves adjusted in vacation it will has some value
	Dept	int	dept id
	RetCode	int	this column has value if employee is returned from vacation
	LV0	float	dynamically created columns based on Leaves which are entered in payroll preferences
	LV1	float	this will maintain number of leaves taken this monthly by a particular employee
Extra table which are used on certain options while calculating payroll			
CxxxForwardAmount			it holds Carry f/w amount information
	MonthlyID	int	has relation with MonthlyId in CxxxMonthlyPayroll
	EmpId	int	employee id
	Amount	float	Carry forwarded Amount
	Location	int	has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. If employee worked in more than one site on a particular month, then this column will maintain location where employee has worked
	Phase	int	this has one to one relation with CxxxEarusg,CxxxDedusg,CxxxPaytable,CxxxLoansusg,CxxxLvsusg. This column will store tag-7 id which will be useful in Region wise posting
	Dept	int	department id
CxxxLeavesAutolncr			This table is used for Auto Increments of Leaves

	MonthlyID	int	has relation with MonthlyId in CxxxMonthlyPayroll
	Empld	int	Employee id
	LeaveType	smallint	Leave Type
	DaysLeft	smallint	Previous Days Left
	DaysWorked	float	Total Days Worked
	LvsIncreased	float	Lvs Increamented

Leaves Application

Leaves application and approval details are maintained in following tables

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxLAPM			Leaves application details
	ApplIndex	int	Primary key
	ApplNo	nvarchar	Voucher number
	EmplID	int	Employee Id
	StDate	datetime	Leave starting date
	EndDate	datetime	Leave Ending Date
	Days	float	Number of days applied for leave
	bVacation	tinyint	if "include authorization in AnnualLeaveApplication" option in PayrollPreferences is selected, then Vacation leaves should be applied from this LeaveApplication module. So this column is used to determine whether current row holds Vacation Leave Application Information or LeaveApplication Information
	DeptHead	int	ignore
	Remarks	nvarchar	reason for applying leave
	ApplDate	int	date on which the leave was applied
CxxxLAPC			
	ApplIndex	int	has one to many relation with ApplIndex of CxxxLAPM
	IncrID	int	just incremental column
	LeaveType	smallint	sequence id of Leaves which are defined in Payroll preferences
	NoofLeavesApplied	float	Number of days applied for leave
CxxxLARM			Leaves approval -1
	ApplIndex	int	has relation with aplindex of CxxxLAPM
	ApprovalIndx	int	primary key
	AprovalNo	nvarchar	Voucher number
	Empld	int	Employee id
	StDate	datetime	Starting date
	EndDate	datetime	Ending date
	Days	float	Number of days approved
	bVacation	tinyint	1=>Annual leave
	bReject	tinyint	1=>leave is Rejected
	VacCode	int	VacCode, if Vacation is processing from Leaves Forms
CxxxLARC			
	ApprovalIndx	int	has relation with ApprovalIndx of CxxxLARM

	IncrID	int	just incremental column
	LeaveType	smallint	sequence id of leaves which are defined in Payroll preferences
	NoofLeavesApproved	float	number of days approved
CxxxLTKM			Leaves Approval-2(final approval)
	ApprovalIdx	int	reference of CxxxLARM table
	Empld	int	Employee id
	StDate	datetime	starting date
	EndDate	datetime	ending date
	Days	float	Days approved
	bReject	tinyint	1=>leave is rejected
CxxxLTKC			
	ApprovalIdx	int	reference of CxxxLTKM
	IncrID	int	Unique Id
	LeaveType	smallint	sequence id of leaves defined in payroll preferences
	NoofLeavesTaken	float	days sanctioned
CxxxLENM			table is used to store encashed leaves details
	ApplIndex	int	
	ApplNo	nvarchar	voucher number
	Empld	int	Employee id
	EncashDate	smallint	Encashment date
CxxxLENC			
	IncrIndex	int	just incremental column
	ApplIndex	int	reference of CxxxLENM
	LeaveType	tinyint	sequence id of leaves which are defined in Payroll preferences
	NoofLeavesEncashed	float	No. of Leaves Encashed
	RatePerDay	float	Rate Per Day
	Amount	float	encashment amount
CxxxLeaveDescription			Calendar for public holidays
	LeaveDate	datetime	Holiday date
	Description	nvarchar	reason for Holiday
	Code	int	Ignore
	iCatId	int	category id. Used in FOW

Advances

Advance to the employee will be give to the employee thru this module

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxAdvancereq			this table will maintain Loan Request Information
	rIndex	int	Unique Key
	RequestId	nvarchar	Voucher Number
	Empld	int	Employee ID
	ReqDate	datetime	Request Date
	Amount	float	Request Amount
	AdvanceCategory	smallint	sequence number of Loan which is entered in PayrollPreferences=>Loans tab
	Reason	nvarchar	reason for taking loan
CxxxAdvancePayment			Loan Approval information
	Code	int	primary key
	Document	nvarchar	document number
	RequestId	nvarchar	refers RequestId Column in CxxxAdvanceReq

	Empld	int	Employee Id
	SanctionDate	datetime	Advance Sanction Date
	DedDate	datetime	Deduction date
	Amount	float	Amount sanctioned + interest Amount
	NoofInstallments	smallint	number of installments
	AdvanceCategory	smallint	sequence id of loan which is entered in Payroll preferences=>Loans tab
	Reason	nvarchar	if the Advance Request is rejected, then this will store reason for rejection
	OpFlag	bit	to denote whether it is forwarded from last year
	DocNumber	nvarchar	Voucher number
	IType	tinyint	0=>None 1=>Flat 2=>Diminishing
	IRate	float	interest rate in %
	LoanAmount	float	Amount sanctioned
	bRejected	int	1=> Loan request is rejected
	strRejReason	varchar	Reason for Rejecting the Loan
	iAuthLvl	int	used by ACK. (loan Approval should be authorized by different HR users. So this column will maintain flag which specifies whether loan is Accepted or rejected
	LoanAmountHOS	float	when there are 2 Level Authorization, then Following columns are used. Data entered in 1'st level (HOS level) will be stored in following columns
	IRateHOS	float	interest rate entered by HOS
	ITypeHOS	int	0=>None 1=>Flat 2=>Diminishing
	NoOfInstHOS	int	number of installments
	iDedDateHOS	int	Deduction date
	bRejectedHOS	int	1=> Loan request is rejected
	strRejReasonHOS	varchar	Reason for Rejecting the Loan
CxxxAdvancePaymentChild			Installment Amounts
	Code	int	reference of CxxxAdvancePayment
	InstDate	datetime	Installment date
	InstAmount	float	Installment Amount
	rIndex	int	just index
	IAmount	float	Interest Amount

Arrears

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxArrearsMain			
	CalcDate	datetime	date on which arrears are getting calculated
	Code	int	primary key
CxxxArrearsChild			
	Code	int	reference of CxxxArrearsMain
	EmpID	int	Employee id
	MonthYear	datetime	month wise arrears values are maintained
	Basic	float	Difference Amount
	OT	float	OT Amount

	HOT	float	HOT Amount
	E0	float	dynamically created columns for storing earnings
	E1	float	
	E2	float	
		
	D0	float	dynamically created columns for storing deductions
	D1	float	

Sector

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxSECM			this will hold details for air ticket cost
	SectorName	nvarchar	Sector Name
	AirLine	nvarchar	air lines name
	SectorID	int	primary key
	UserAuthFlag	int	used by ACK. Sector master details will be authorized by different users. Appropriate bit will be set if user authorizes this information
	UserRejFlag	int	used by ACK. Appropriate bit will be set if user rejects it
CxxxSectFares			
	SectorID	int	reference of CxxxSeCM
	SeasonName	nvarchar	Name
	SeasonDate	datetime	effective date
	EcoFare	float	cost for Economic class
	BusinessFare	float	cost for business class
	FirstClassFare	float	cost for first class
	IncrIndex	int	incremental column
	EcoFare_Child	float	economic class (for children)
	BusinessFare_Child	float	Business class (for children)
	FirstClassFare_Child	float	First class(for children)
	EcoFare_Infant	float	Economic fair(for infants)
	BusinessFare_Infant	float	business fair(for infants)
	FirstClassFare_Infant	float	First-class fair(for infants)

Shifts

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxShiftMaster			
	ShiftID	int	primary key
	ShiftName	nvarchar	Shift Name
	TimeIn1	int	Time In
	TimeIn2	int	in from break/lunch
	TimeOut1	int	Timeout
	TimeOut2	int	out for break/lunch
	HalfdayTimeOut	int	Time out on Half Day
	OffDay1	smallint	weekly off1 (Sunday-1 mon-2..)
	OffDay2	smallint	weekly off2
	HalfDay	smallint	halfday
	LateMark1	smallint	grace minutes for late mark
	LateMark2	smallint	grace minutes for late mark
	Allowance	float	Shift Allowances
	EarlyGrace	float	grace minutes for early leaving
	DinnerTime	int	Dinner Time for shift

	DinnerAllow	float	Dinner Allowance
	TeaTime	int	Tea Time
	TeaAllow	float	Tea Allowance
	TeaTime2	int	Second Tea Time
	TeaAllow2	float	Second Tea Allowance
	OTTime	int	OT Time
	MinWorkingHrs	float	Minimum Hours to work for OT
	MinHalfHrs	float	Minimum Hours for Half Day
	MinFullHrs	float	Minimum Hours for Full Day
	MaxOTHrs	float	Maximum OT Hours
	OTNotApplied	int	OT Applicable or not
	EarlyLoginGrace	int	should not login before this time to shift
	MinOTHrs	float	Minimum OT Hours
	OTTime2	int	Second OT Time
	AbscentGrace	float	Grace time for Absent
	fMinHrPerWeek	float	min's hrs per week. If hr's per week exceeds this value, then Remaining extra hr's will be considered as OT hrs
CxxxShiftChange			when shift is assigned from "shift change" screen, the information will be stored in this table
	Date	smallint	date in int format on which shift is assigned, it will continue till next shift is assigned on other date
	EmpID	int	employee id
	NewShift	smallint	shift id
	bCalcOt	int	flag to calculate OT or not
	ToDate	int	Max Date for allocated shift
	iIndex	int	Unique Key

Time-Sheet

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxSwip			data which is entered in "Swipe card" or "Swipe card grid" screen will be stored in this table
	SwipeDate	smallint	Date In
	SwipeTime	int	IN Time
	OutDate	smallint	Date Out
	OutTime	int	OUT Time
	EmpId	int	Employee ID
	Code	int	Unique Key
	Location	int	Location worked
	BreakOut	int	Break Time Out
	BreakIn	int	Break Time In
	Shift	int	shiftid
	Edited	int	flag to know whether edited or directly imported
	bCalcOT	int	for amaintit
	TEMP_DEF_DEPT	int	for amaintit
	SwipeCode	int	this is used while importing swipe information from Text or MDB file. Depending on Swipe code, import Module will decide whether the time is IN-Time or OUT-time
	SwipeInfo	int	this is used while importing swipe information from Text or MDB file.
	OnDutyHrs	float	OnDuty Hours

	PermissionHrs	float	No. of hours for which he/she has taken permission
	OfficialHrs	float	No. of Official Hours
	PersonalHrs	float	No. of Personal Hours
	Remarks	varchar	Remarks

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxDailyPayroll			data which is entered in "Daily payroll" screen will be stored in this table
	rIndex	int	ignore just incremental index
	Empld	int	Employee id
	Dailypaydate	datetime	date on which daily payroll is calculated
	Locations	int	if employee works in more than 1 location on a particular day, then Location id will be stored in this column
	Status	float	1-present 2-halfdaypresent 3-leave 4-halfdayleave 5-absent 6-holiday 7-halfdayleavehalfdaypresent. if it is hourly Payroll, then this column will store hours worked
	LeaveType	int	sequence of leave in leaves order in payroll preferences. If employee is on leave, then this column will store Leave Sequence id.
	OtHours	float	Over time hours worked
	HolidayOT	float	holiday overtime.
	Late1	int	late minutes
	Late2	int	Late minutes after lunch time
	Category	int	CategoryID
	Dept	int	department id
	Shift	int	this will store shift id which is assigned to this employee
	Phase	int	tag7 id.
	Remarks	nvarchar	brief description
	DCode	int	unique code
	OffShore	float	this is used for VMGL. same as status column.
	OffOT	float	this is used for VMGL. same as OTHours column
	OffHOT	float	this is used for VMGL. same as HolidayOT column.
	ExtraHrs	float	Extra Hours
	PreLateMin	float	No. of minutes between Late mark1 and Employee Time In
	Designation	nvarchar	Designation
	szDocNumber	varchar	Voucher number
	DL0	float	dynamically created columns for storing daily inputs. Daily Input formula which is entered in Payroll preferences will be evaluated and stored in this column.
	IDL0	float	dynamically created columns for storing daily inputs. This will hold value which is entered in dailyInput column.
CxxxDlyBreaks			If employee is going out during duty hours his Out and In times are maintained in this table

	EmpID	int	Employee Id
	OutDate	int	date on which employee took break
	OutTime	int	Out Time
	InTime	int	Out Date
	InDate	int	In Date
	Reason	nvarchar	reason for the break
	Official	int	1-if official 0-if personal

Vacation

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxVacationInfo			This table is used when employee is going for vacation
	DocNumber	nvarchar	voucher number
	Code	int	primary key
	EmpId	int	employee id
	StDate	datetime	vacation starting date
	EndDate	datetime	vacation end date
	LeavesUsed	float	Number of Vacation Leaves Used
	LeavesWoPay	float	Unpaid Leaves
	Net	float	Vacation Amount
	bEncash	tinyint	1=>encash vacation 0=>otherwise
	LeavesEligible	float	number of eligible vacation leaves
	PHoliDays	int	No. of public holidays
	LvsAccured	float	Total Leaves Accrued
	WeeklyOffs	int	No. of weekly offs
	CalcDate	datetime	Date on which vacation information is entered.
	LeavesBf	float	Leaves b/f
	LeavesBfOff	float	offshore leaves b/f
	TotalTickets	float	number of eligible Air tickets.
	TicketsUsed	float	number of air tickets used for this vacation
	OffDaysWorked	float	offshore days worked
	OffLvsEligible	float	offshore leaves eligible
	OffTicketsEligible	float	offshore Tickets eligible
	DaysWorked	float	total number of days worked (work experience)
	PHWorked	float	No. of public holidays worked
	WWorked	float	No. of weekly offs worked
	OffPHWorked	float	No. of public holidays worked in the offshore
	OffWWorked	float	No. of weekly offs worked in the offshore
	ReturnDate	datetime	return date from the vacation
	VacEB	float	earned basic for vacation
	bFinal	int	0=>Vacation details 1=>Final settlement details
	JourneyDate	datetime	date on which employee is leaving for vacation
	LvsEncashed	float	Lvs Encashed
	Remarks	nvarchar	Remarks
	bNoCalcVac	int	flag whether to calculate vacation or not
	CONT_SRVC_UPTO	datetime	Amaintit specific
	AdjustLvs	float	leaves adjusted
	AckPays	int	used by ACK
	AirTktEncash	int	air ticket encashed or not
	iPayVacAmtDate	int	this will store Month Value. Vacation Amount will be included when Payroll is

			calculated for this month
	M0	float	dynamically created columns for each vacation input created in vacation preferences
	M1	float	this will store Vacation Input Values which is entered in Vacation Leave Application or Final Settlement Screen
	...		
	VE0		dynamically created columns for each vacation earning created in vacation preferences
	VE1		vacation earning value
	...		
	VD0		dynamically created columns for each vacation deduction created in vacation preferences
	VD1		vacation deduction Value
	...		
CxxxVacRet			this table is used when employee returns back from vacation
	RetCode	int	primary key
	VacCode	int	reference of CxxxVacationInfo table
	ReturnDate	datetime	return date
	LvsAdjusted	float	if Employee Returns late from Vacation, then Late days which is adjusted from Vacation Eligible Days will be stored in this column.
	Net	float	Net on vacation period
	VacRetEB	float	Earned basic on vacation period
	LvsAccured	float	No. of leaves accrued in the vacation
	VE0	float	dynamically created columns for vacation earnings created in vacation preferences
	VE1	float	Vacation earning value
	...		
	VD0	float	dynamically created columns for vacation Deduction created in vacation preferences
	VD1	float	Vacation deduction value
	...		
CxxxVacationSplit			This table is used if employee is going for vacation in middle of month and wants to make salary payment with vacation amount
	Code	int	reference of CxxxVacationInfo table
	StDate	datetime	Starting Date
	EndDate	datetime	Ending Date
	NoofDays	float	No. of Days
	Basic	float	basic
	OTHrs	float	OT hours
	HoTHrs	float	hot hours
	OT	float	OT
	HoT	float	HOT
	OffHrs	float	offshore hours

	OffOTHrs	float	offshore OT hours
	OffHoTHrs	float	offshore HOT hours
	Location	int	location id
	Phase	int	phase id
	E0	float	dynamically created columns for each earning created in payroll preferences
	E1	float	
		
	D0	float	dynamically created columns for each deduction created in payroll preferences
	D1	float	
		
	ML0	float	dynamically created column for each monthly input defined in payroll preferences
	ML1	float	
	..		

Training

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxTrainingMaster			
	TrainingCourseId	int	primary key
	TrainingCourseNo	nvarchar	
	StDate	smallint	Training start Date
	EndDate	smallint	Training End Date
	StTime	int	Training Start Time
	EndTime	int	End Time
	Description	nvarchar	training course description
	Cost	float	training cost
	FID	int	Biodata format ID which gets created in CxxxBiodataStructures table
	Path	varchar	used by ACK. This will store Path of the which contains details of the course

Achievements

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxAchievements	simple table which has no relation with other table		
	EntryNumber	nvarchar	Entry number
	EmpID	int	Employee Id
	Date	smallint	Achievement/drawback Date
	Type	nvarchar	"A"=>Achievement "D"=> drawback "W"=>Warning
	Description	nvarchar	remarks
	Weightage	int	weightage marks
	rIndex	int	unique id
	szRefNo	varchar	Reference Number if any

Location change

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxLocationChange	simple table which has no relation with any other tables		
	RequestId	nvarchar	Request number

	EmpID	int	Employee Id
	Date	smallint	Date
	Time	int	Time
	NewLocation	smallint	New Location

Payroll External

Payroll External Modules (External modules in Misc.dll)

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxGradingMaster			
	strGrade	char(10)	Grade
	dPercentage	float	percentage of increment
	iYear	integer	year
CxxxBonus			
	empid	int	empid
	performance	float	performance
	Yr	int	year
	DateofCalc	int	Date of Calculation
	Cat	int	Category
	BonusAmt	float	Bonus Amount
CxxxIncr			
	empid	int	empid
	DateofRev	int	Date of Revision
CxxxPromotion			
	empid	int	empid
	DateofRev	int	Date of Revision
	Cat	int	Category
	Amt	float	amount
CxxxMissionHrsReq			
	DocNo	varchar	unique id
	RDate	datetime	request date
	EmpID	int	empid
	Mission	varchar	mission name
	tmFrom	int	minimum time
	tmTo	int	maximum time
CxxxMissionHrsApr			
	DocNo	varchar	unique id
	ADate	datetime	approval date
	EmpID	int	empid
	Mission	varchar	mission name
	Status	int	approval status, 0 - rejected and 1 - approved
	tmFrom	int	minimum time
	tmTo	int	maximum time
CxxxLvsReq			
	DocNo	varchar	unique id
	EmpId	int	empid
	LType	int	leave type
	LDate	int	leave date
	tmFrom	int	minimum time
	tmTo	int	maximum time
CxxxLvsApr			
	DocNo	varchar	unique id

	Empld	int	empid
	LType	int	leave type
	LDate	int	leave date
	tmFrom	int	minimum time
	tmTo	int	maximum time
	Status	int	approval status, 0 - rejected and 1 - approved
	GMStatus	int	GM approval status, 0 - rejected and 1 - approved
CxxxAttInfo			
	Empld	int	empid
	LvsCF	float	No. of leave carry forwarded
	WorkedHrs	float	No. of hours worked
	ActHrs	float	Actual Hours
	ADate	datetime	Approval Date
CxxxTaxSlabs			
	Code	int	unique id
	Year	int	year
	Slab	float	slab
	TaxRatio	float	Tax Ratio
	FixedTax	float	Fixed Tax Amount
CxxxTaxAmt			
	Code	int	unique id
	EmplID	int	empid
	Tax	float	Tax Amount
	TCDate	int	Tax Calculation Date
CxxxSwipeCodes			this table will maintain swipe code information which is used to distinguish timeIn and Time out while importing Swipe details from File
	Code	int	unique id
	SwipeIn	varchar	IN code
	SwipeOut	varchar	OUT code
	Descr	varchar	description
CxxxLeave			
	Empld	int	empid
	FromDate	datetime	From Date
	ToDate	datetime	To Date
	Returned		1-returned, 0 - not returned
	ReturnDate	datetime	Return Date
CxxxPayUsgLog		this table is used to maintain usage log information	
	id	int	primary key
	luser	varchar	User name
	ADate	int	Date
	ATimein	int	From Time
	ATimeout	int	To Time
	Activity	varchar	Activity preformed
CxxTrainingMaster1			
	iTrainingId	int	primary key
	szTrainingName	varchar	training course Name
	szPurpose	varchar	Purpose of this training
CxxxTrainingMasterBody1			
	ild	int	Id
	iHeaderId	int	this refers to iTrainingId column in CxxxTrainingMaster1 table.
	iCategory	int	categoryId for which this Training is

			applicable
CxxxTrainer			Trainer details
	iTrainerId	int	primary key
	szName	varchar	Name of the trainor
	szQualification	varchar	trainer's Qualifications
	szAddress	varchar	Address
	szCity	varchar	City
	szPin	varchar	Pin
	szState	varchar	state
	szPhone	varchar	Phone number
	szEmail	varchar	Email id
CxxxTrainorCost			this table will store charges taken by Trainor to give particular training
	ild	int	primary key
	szDocNo	varchar	voucher number
	iTrainerId	int	this refers to ITrainerId column in CxxxTrainer table
	iTrainingId	int	this refers to iTrainingId column in CxxxTrainingMaster1 table.
	dCost	float	Training Cost
	iCurrencyId	int	Currency
	iUnit	int	0=>Hour, 1=>day, 2=>Program
CxxxTrainingReqHeader			Training Request
	ild	int	primary key
	iType	tinyint	
	szReqNo	varchar	voucher number
	iCategory	int	Category id
	iNoDays	smallint	Number of days
	iStartDate	smallint	Starting date
	iEndDate	smallint	Ending date
	iStartTime	int	training Start time
	iEndTime	int	Training End Time
	iTrainingId	int	this refers to iTrainingId column in CxxxTrainingMaster1 table.
	iTrainerId	int	this refers to ITrainerId column in CxxxTrainer table
	iTrainingType	tinyint	0=>Internal 1=>External
	iTrainerType	tinyint	0=>Internal 1=>External
	iEnteredBy	int	User id who entered this Request
CxxxTrainingReqBody			
	ild	int	primary key
	iHeaderId	int	this refers to ild column in CxxxTrainingReqHeader
	iEmpId	int	Employee who wishes to take this training
	szRemarks	varchar	brief description
CxxxFeedback			Feedback of training
	ild	int	primary key
	iHeaderId	int	this refers to ild column in CxxxTrainingReqBody
	iEmpId	int	Employee id
	szRemarks	varchar	Remarks
CxxxAirFares			
	rindex	int	unique id
	iCat	int	category id
	iPayType	int	Payment Type

	iEntitlementPeriod	int	Entitlement Period
	iChildCnt	int	Child Count
	iChildAge	int	Child Age
	iSectorId	int	Sector Id
	iCurrency	int	Currency Id
	iChildAgeUB	int	Max Child Age
	iChildAgeLB	int	Minimum Child Age
	fAmt	float	Amount for each air ticket
	fChildAmtUB	float	Maximum Child Amount
	fChildAmtLB	float	Minimum Child Amount
	fChildAmt	float	Child Amount if there is no age slabs
	blsSpouseElig	bit	1-spouse eligible for air ticket, 0 - spouse not eligible for air ticket
	bAgeRestriction	bit	1-age restriction on child age,0-no age restriction
	iTravelClass	int	Travel class
	iCat1Share	int	first slab air ticket amount for child
	iCat2Share	int	second slab air ticket amount for child
	iCat3Share	int	third slab air ticket amount for child
	iCat1Age	int	first Slab max child age
	iCat2Age	int	second slab max child age
	iCat3Age	int	third slab max child age
	bEldrChd	bit	1-consider elder child first,2-consider older child first to calculate air ticket
CxxxOverTime			Request for overtime
	rindex	int	primary key
	szDocNo	varchar	voucher number
	iDate	int	Date
	iEmpld	int	Employee id
	iDepld	int	Department
	szOrganisation	varchar	Organization Name
	iStartTime	int	Start time
	iEndTime	int	End Time
	iReason	int	Reason for OT
CxxxOverTimeAppr			Over time Approval
	ild	int	primary key
	iReqlId	int	this refers to rIndex column in CxxxOverTime
	iStartTime	int	Start time
	iEndTime	int	End Time
	bApproved	bit	1=>Approved , 0=>Rejected
CxxxDesignation			Designation master
	ild	int	primary key
	name	varchar	Designation
CxxxOTPref			Overtime Preferences. this is used by CBK .
	id	int	primary key
	value	float	OT Amount
	descr	varchar	OT description
CxxxOverTimeCBK			
	id	int	unique id
	MonthId	int	Monthly id . This will refer to MonthlyId in CxxxMonthlyPayroll table
	iEmpld	int	Employee id

	NormalOT	float	Normal OT Hours
	FridayOT	float	Friday OT Hours
	HOT	float	Holiday OT Hours
	SpecialOT	float	Special OT Hours
	OTRate	float	OT Rate
	SpecialOTRate	float	Special OT Rate
	NormalOtRate	float	Normal OT Rate
	FridayOTRate	float	Friday OT Rate
	HOTRate	float	Holiday OT Rate
	TotOtAmt	float	Total OT Amt
	iAdviceDate	int	Advice date
	iDate	int	Date
	iRevisedId	int	Revise Id
	TotalOTAmt	float	Total OT Amount
	TotalHotAmt	float	Total HOT Amount
CxxxLoansSettlement			if Employee Pays loan Amount through cash, then this table will record all these information. It will be useful while calculating Earned Basic and net Amount
	id	int	primary key
	empld	int	Employee id
	monthlyId	int	Monthly id . This will refer to MonthlyId in CxxxMonthlyPayroll table
	iDate	int	Date on which money is paid
	LN0	float	dynamically created columns based on Loans which are entered in payroll preferences
	LN1	float	this will store amount paid by the employee
CxxxNewMonthlyInput			
	id	int	primary key
	iEmpld	int	Employee id
	iDate	int	date
	Ear0	float	dynamically created columns based on Earnings which are entered in payroll preferences
	Ear1	float	this will store Earning value
	...		
	Ded0	float	dynamically created columns based on Deductions which are entered in payroll preferences
	Ded1	float	this will store Deduction Value
CxxxTermination			For Termination/Internal transfer details
	id	int	primary key
	iType	int	2- for Internal transfer. Otherwise Termination
	szDoc	varchar	voucher number
	iEmpld	int	Employee id
	iEffectiveDate	int	Effective date
	iLastDate	int	Last Day in position
	iNoticePeriod	int	Notice Period
	bReplacement	bit	1=>Replacement necessary, 0=>Replacement not needed
	szReason	varchar	reason for status change

	szPosition	varchar	current designation
	szNewPosition	varchar	New Designation

External Modules in Import.dll

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxMapTable			
	ReqNo	nvarchar	unique id
	FormatID	int	format id
	HrFld	nvarchar	Database Field
	EmpFld	nvarchar	Focus Fields
	FldType	int	field type
	ValueType	int	value type
	Path	nvarchar	path of the file from which data is imported
CxxxTxtCfgMaster			
	iCode	int	unique id
	strTxtFilePath	varchar	File path
	iType	int	1-multiple swipe per day,2-single swipe per day
	iDelimiter	int	delimiter
	iDateFmt	int	date format
	iChangeSwipln	int	assign swipe out to swipe in, if swipe is not available
CxxxTxtCfgChild			
	iCode	int	unique id
	strFocusCol	nvarchar	focus column
	iSize	int	size of field
	ilgnore	int	1 - ignore the field to import
	iSNo	int	serial number
	strCondVal	varchar	condition to import
CxxxSwipeTmp			this table is used while importing swipe information from a File
	iCode	int	
	iLocation	int	location id
	iEmpId	int	Employee id
	iDate	int	Date
	iTime	int	Time
	strSwipeType	varchar	swipe code
	iShift	int	shift id
	SwipeInfo	int	
CxxxMonthlyPayCfgMaster			when Attendance or swipe information is imported , then the setting will be stored in this table
	iCode	int	
	strDBFFilePath	varchar	File path
	strType	int	1-multiple swipe per day,2-single swipe per day
	iConnType	int	1-odbc, 2 - ado
	iChangeSwipln	int	assign swipe out to swipe in, if swipe is not available
	iEmpCodeSize	int	Maximum digits to consider for

			employee code
CxxxMonthlyPayCfgChild			
	iCode	int	this refers to iCode column in CxxxMonthlyPayCfgMaster table
	strFocusColumnName	varchar	Focus6 field
	strDBFColumnName	varchar	Database field
CxxxLvslImport			
	Empld	int	empid
	dateofrev	int	month end date
	LV0	float	Dynamically created columns
	LV1		

Recruitment

RECRUITMENT DATABASE STRUCTURES

TABLE NAME	COLUMN NAME	DATATYPE	REMARKS
CxxxBioDataStructures			This table is used to define field definitions for biodata from fields
	Alias	nvarchar	display name in form
	BioDataFormatID	int	internal format id
	BioDataName	nvarchar	Name of biodata structure
	BName	nvarchar	DB field name
	cdIndex	int	incremental column
	DisplayType	tinyint	specifies display type of field like text field, Radio Button, Selection. 7 to 16=>Tags, 1=>Text, 30=>Link, 2=>Radio,3=>Checkbox, 4=>Date control, 5=>Time Control, 17=>Farsi Type, 6=>selection
	Flags	int	if Flags&0x1>0 then it is "read only". If Flags&0x2>0 then it is "Internal field". If Flags&0x4>0 then it is "Value Mandatory". If Flags&0x8>0 then it is "Marking field". If Flag&0x10>0 then it is "Interview Marking". If Flag&0x20>0 then it is Specification Field. if Flag&0x40>0 then it is "Authorization Field". if Flag&0x80>0 then it is "Joining Field"
	FldPos	int	Position of field
	GroupName	nvarchar	Tab names
	Length	int	length of field in database columns
	Max1	int	ignore
	Max2	int	ignore
	Max3	int	ignore
	Min1	int	ignore
	Min2	int	ignore
	Min3	int	ignore
	NoofDecimals	int	if field type is float, then no of decimals
	SelectionValues	nvarchar	if field type is selection, then this column will hold selection Items with comma separated

	Type	tinyint	type of control to be displayed. 7-Tag,3-Float, 2-Number, 1-text, 4=yesNo, 5-datetime, 17-FarsiDateTime, 6=>selection
	OtherFormat	tinyint	0=recruitment, 1=Appraisal, 2=Training, 3=Exit Interview
	DefaultFormat	tinyint	1=>if current format is a Default format
	szUserId	varchar	used by ACK.
CxxxBn ,Ex:C0101B2			here in is format id. For each biodata Format, one separate table will be created with Name CxxxBn where xxx=>company prefix, n=>FormatId. This is used in Biodata entry screen
	ReqID	int	will refer cdIndex column in CxxxReqRecr
	BioDataId	int	id
	BioDataCode	nvarchar	bio data code
	Flags	int	
	AuthFlag	int	0-dept head approval,1 user approval
	userAuthFlg	int	approved
	userRejFlag	int	not approved
	strPath	varchar	bio data path
			Remaining columns are dynamically created depending on values that are entered in Biodata definition screen
C020OFORMATS			used in Biodata entry screen
	EmpID	int	Employee id
	TransDate	int	Biodata entry date
	FID	int	Format Id which will refer BioDataFormatID column in CxxxBioDataStructures table
	BID	int	refers BioDataId column in CxxxBn
CxxxReqRecr			request for recruitment
	ReqNo	nvarchar	request number
	BioDataFormatId	int	Format Id which will refer BioDataFormatID column in CxxxBioDataStructures table
	Department	int	dept id for which recruitment process is going
	Category	int	-
	Designation	nvarchar	position for which recruitment is going
	NoOfEmployees	int	number of employees needed
	BioDataID	int	refernceid in C050B2 table. If there are any specification fields, then the values entered in specification fields will be first stored in CxxxBn table, and this column will hold BiodataId
	cdIndex	int	incremental column
	EnteredBy	nvarchar	user name
	AuthFlag	int	flags for authorization
	RejectFlag	int	flags used to indicate that biodata is

			rejects
	ReqDate	int	request date
CxxxMapTable	this table is used to map fields from recruitment to employee master		
	ReqNo	nvarchar	Biodata format name
	FormatID	int	biodata format id
	HrFld	nvarchar	Name of Recruitment field
	EmpFld	nvarchar	name employee master field
	FldType	int	type of field like int,date,char
	ValueType	int	always -1 for recruitment
	Path	nvarchar	not required for recruitment mapping but it is used when it is mapped from excel file to import data to employee master, then it is the path of excel file

API - Introduction

This document describes the features of the Focus RT COM API which allows third party applications to query information from and send commands to Focus RT Data tables. Through FOCUS API, applications can query the Masters and Transactions tables and send information to the tables containing the masters and transaction data. Apart from being able to query and send information, Focus RT built-in functionality allows the external modules written using API to be triggered on events within Focus RT like on Save, On Delete, On Update and so on.

This COM API documentation package includes:

- This documentation
- FocusAPI.Doc - the original document on classes, field list and documentation.

COM Registration

Before you can interface a 3rd party application with Focus RT, you must install Focus RT on the computer where you intend to run the 3rd party application. The COM interface is registered as part of the Focus RT installation process.

If, for some reason, you need to register the COM API again, execute option.ocx, company.ocx and FocusApi.Ocx with "/RegServer" as a command line argument. For instance, assuming these files are located in C:\FocusRt, use the following command line to register Focus APIs:

```
"C:\FocusRT\company.ocx" /RegServer
```

```
"C:\FocusRT\options.ocx" /RegServer
```

```
"C:\FocusRT\FocusApi.ocx" /RegServer
```

To unregister the Focus RT API, use the following command:

"c:\FocusRT\Company.ocx" /UnregServer

"c:\FocusRT\option.ocx" /UnregServer

"c:\FocusRT\FocusApi.ocx" /UnregServer

Note: You will be able to register the APIs only if you have registered the inf for Focus and also if the registered CDID of Focus RT include API component. Those who have not purchase the API will not be able to register and use the API's though the files may exist

Object/Classes	Description
<u>Company.ocx</u>	Fetches information related to Company details.
<u>Option.ocx</u>	This object consists of same behavior as a combo box control with drag and drop property. Can be used to populate option box with focus master / Tag list . Following are the list of properties at design and run time .
<u>FMASTER</u>	This class is used to Create, Read,Edit,Delete Masters like Accounts, Products, tags.
<u>FTRANSACTION</u>	This class is used to Create new documents in existing document types, Read from records from existing document types, update documents or delete documents
<u>FREPORT</u>	This class is for accessing data records to read transactions for reporting. The class members include accounts data, products data, account and product balances....
<u>FMASTERREPORT</u>	This class is to read and report on masters.
<u>FLINKS</u>	This class fetches the pending Linked documents
<u>REFERENCE</u>	This class would access and fetch pending bill references of customer and vendors for AR/AP module.
<u>FUNITS</u>	This class would read/write to units from product master.
<u>FCURRENCIES</u>	This class would read/write to currency master
<u>FRATEMASTER</u>	This class would read/write rates in product master
<u>FFIXEDASSET</u>	This class reads/writes data from/to fixed assets module.
<u>FPAYROLL</u>	This class reads/writes data from/to payroll module.
ProductionBatchList	Read/Write to Production Batches for manufacturing Module
FProduction	Read/Write to Production BOM
<u>Logins</u>	Read/write security users
<u>FRemoteExec</u>	Class available in FocusApi to connect without a DSN and database password

in the Focus RT folder.

CLASS LIST

Company.ocx

This object also behaves as a combo box control but can be attached to get data for available list of Focus companies. Provides following information as

Field List

Properties	Explanation	Parameter/ Return Type
AccountingDate	To Get / Set Returns accounting date for selected company	Date
CompanyCode	To Get / Set Returns code for selected company	String
CompanyName	To Get / Set Returns code for selected company	String
InitCompany	To initialize required company ,Need to pass that comp code in form of integer	Long
InitializeSelectedCompany	To initialize selected company .	Pass 0
Close	Close and set back memory from this control	
CompanyDetails		
Open	To open the current selected company	0
CompanyName	To Get / Set Name of company	String
Address1	To Get / Set address of company	String
Address2		String
Save	Saving all settings for current company	
Close	Close objects	Int
Remaining properties are identical other com objects .		

Sample Code

To save New Company details
`Dim cd As New CompanyDetails`

```

cd.New
cd.CompanyName = TxtCName.Text
cd.Address1 = txtAdd1.Text
cd.Address2 = txtAdd2.Text
cd.AccountingDate = MskAccDate.Text
cd.SecurityLevel = 0
cd.Save

```

To Initialize and get company list

```

CompanyList.InitializeCompany TxtCCode.Text
TxtCName.Text = CompanyList.CompanyName
MskAccDate.Text = CompanyList.AccountingDate

```

Option.ocx

This object consists of same behavior as a combo box control with drag and drop property. Can be used to populate option box with focus master / Tag list . Following are the list of properties at design and run time .

Field List

Properties	Explanation	Parameter/ Return Type
CurSelId	To Get / Set Returns masterid for selected master type	Long
SelName	To Get / Set Returns name for selected master type	String
RestrictToAGroup	To restrict option box to a group	Long
SelAlias	To Get / Set Returns alias for selected master type	String
SelCode	To Get / Set Returns code for selected master type	String
Initialize	To initialize a comp instance	Integer
Account Type	Set/ Get account type (Purchase account,Sales or any other account type listed in account master)	Integer
SelectionType	Set selection type property to load option box with master/ group or leaf .	Integer
Remaining properties are identical other com objects .		

Sample Codes

Option box name as = Optcustacc
Getting name and other detail from it

```

Dim strCode As String
Dim strName As String
Dim varcustId As Long

strName = optcustacc.SelName

```

```
strCode = optcustacc.SelCode
varcustId = optcustacc.CurSelId
```

Updated selected account master

```
Seqid = fm.OpenMaster("Account", "Name", optcustName)
if Seqid > 0 Then
    fm.SetField "Name", Trim(txtcustname.Text)
    fm.SetField "Type", "3"
    fm.SetField "Address1", Trim(txtcustadd.Text)
    fm.SetField "Tel", Trim(txtcusttel.Text)
    fm.SetField "P.B. No", Trim(txtcustpbno.Text)
    fm.SetField "Off No", Trim(txtcustoffno.Text)
    fm.SetField "Mobile", Trim(txtmobno.Text)
    fm.SetField "email", Trim(txtemail.Text)
    fm.SetField "Emirates", Trim(txtemail.Text)
    Seqid = fm.Save
    fm.Close
    If Seqid > 1 Then
        MsgBox "Customer A/c Created Successfully", vbInformation + vbOKOnly
        optcustacc.SetFocus
```

Restrict option

optcustacc.RestrictToAGroup CLng(varMid)

Purpose: While loading any master it restricts to a given group id.

Input Parm: Sequence Id of the group

Return Value:

Success returns 1

Failure returns 0

SAMPLE CODE FOR CONNECTION STRING

```
Public Sub Connection()
    DsnNameReg = "Focus6"
    PWordReg = "focus"
    cop = Cd.Code

    On Error GoTo ErrTrap
    If conn.State = 1 Then conn.Close
    conn.Open "Provider=MSDASQL;Data Source= " & DsnNameReg & ";Initial
    Catalog=Focus5" & cop & " ;", UNameReg, PWordReg
    Exit Sub
ErrTrap:

    If Err.Number = 364 Then
        Exit Sub
    ElseIf Err.Number = -2147467259 Then
        MsgBox "Connection Could Not Be Resolved", vbInformation, "Focus"
    End If
End Sub
```

FMaster

The FMaster Class provides functions to Create , Read, Edit and Delete Masters in Focus.

Methods

New (MasterType as String) As Long

Purpose: Creates a new Master .

InputParams:

MasterType: Master Name Ex: Account , Product, Tag Names

Return Value:

On Success returns MasterId.

On Failure returns -1

Masterid 0 – account , 1-product

OpenMaster(MasterType as String, SearchBy as String, Value as String) as Long

Purpose: Opens an existing Master to Read , Edit

InputParams :

MasterType : Master Name Ex: Account , Product, Tag Names

SearchBy : this param value can be "seqId" or "name" or "code" or "alias".

Value : Value of SearchBy Field.

Return Value: On Success returns Opened Master's SeqId
0 : Failure

SetField (Name as String, Value as String) as Long

Purpose: Sets values of a specified field

Input Params:

Name: Field Name

Value: Field Value

GetField(Name as Sting , Value as String) as Long

Purpose: Gets value of a specified field

InputParams:

Name : Field Name

Return Value:

Value :Field Value

Save() as Long

Purpose: Saves a master

InputParams: -None-

Return Value:

On Success Returns SeqId of Saved Master.
Else Returns 0

Close

Purpose: Closes the Opened master.

InputParam: -None-

Return Value : -None-

IsChildOf (SearchBy as String, Value as String) as Integer

Purpose : To find out whether Current Master IsChildOf Specified Master.

InputParams:

SearchBy : This param value can be "seqId" or "name" or "code" or "alias".

Value : Value of SearchBy Field.

Return Value:

If true returns 1
Else returns 0

Direct Methods**Delete(MasterType as String, SearchBy as String , Value as String , ShowMsg as Boolean) as Long**

Purpose: Deletes a Master

InputParam :

MasterType: Master Name Ex: Account , Product, Tag Names

SearchBy: this param value can be "seqId" or "name" or "code" or "alias".

Value: Value of SearchBy Field.

ShowMsg: if true Confirmation Message will be displayed**Return Value:**

On Success returns 1
0: Failure

GetValue (MasterType as String, SeqID as Long,RetField as Integer, Value as String) as Long

Purpose: To get name or code or alias by passing Seqid

InputParams:

MasterType : Master Name Ex: Account , Product, Tag Names

SeqId: Sequence Id of master

RetField: 0-Name, 1-Code,2-Alias

OutputParams:

Value : requested value (name or code or alias) will be returned in this param.

GetSeqId (MasterType as String , SearchBy as Integer, SearchValue as String)as Long

Purpose: To get SeqId from name or code or alias

InputParams:

MasterType : Master Name Ex: Account , Product, Tag Names

SearchBy : 0-Name , 1-Code, 2-Alias

SearchValue : Name or Code or Alias of a master

SearchBy: Name : Mahmood , swamy. Code : 1,2 Alias : Skay

Field List

Field	Description	Remarks	Set	Get
Seqid	Auto number generated by focus		N	Y
Name			Y	Y
Code			Y	Y
Alias			Y	Y
Parent seqId			Y	Y
Parent Name			Y	Y
Parent Code			Y	Y
Parent Alias			Y	Y
IsGroup			Y	Y
Level			N	Y
Limit			Y	Y
Limit2			Y	Y
CreditDays			Y	Y
Type		0-Cash, 1-Bank, 2-employee, 3-customer, 4-Asset/Liability, 5-Sales, 6-Purchases, 7-Income-expences, 8-FiexedAssets	Y	Y
\$NotePad			Y	Y

ExtraFields can be accessed through their names (Case Sensitive)

Sample Code

Creating Item masters

```
Strseqid = fm.OpenMaster("Product", "Name", "Profiles")
If Strseqid = 0 Then
```

```

fm.New "Product"
fm.SetField "Name", "Profiles"
fm.SetField "Code", "Profiles"
fm.SetField "IsGroup", True
fm.SetField "Type", "3"

seqid = fm.Save
Strseqid = seqid
Else
    Strseqid = fm.GetSeqId("Product", 0, "Profiles")
End If
fm.Close
Set fm = Nothing

```

FTransaction

Methods

NewDocument(VoucherType as String, VoucherNo as String)as Long

Purpose: To create a New Document. Ex:Sales invoice,Receipt..

Input Params:

VoucherType : VoucherType is abbreviation defined in focus for a Voucher. As an example, for sales invoice Voucher Type is "Sal"
VoucherNo: An alphanumeric String. In capitals Pass "" for Next VoucherNo for this VoucherType

ReturnValue:

- 1 - Success
- 0 - Failure (Given Voucher Type is invalid)

LoadDocument(VoucherType as String, VoucherNo as String) as Long

Purpose: to Load an Existing Document. Ex:Sales invoice,Receipt..

Input Params:

VoucherType : VoucherType is abbreviation defined in focus for a Voucher. For example, for sales invoice Voucher Type is "Sal"
VoucherNo: An alphanumeric String.

ReturnValue:

- On Success returns No Of Rows in the Opened Documents.
- 0 - Failure (Given Voucher type is invalid or Voucher No does not exist)

SaveDocument() as Long

Purpose: to Save a Document.

Input Params: None

Return Value:

1-Success
0-Failed

SetField(Name as String , Value as String) as Long

Purpose: to Set value of a Field

Input Params:

Name: Field Name
Value: Field Value

GetField(Name as String , Value as String) as Long

Purpose: to Get value of a Field

Input Params:

Name: Field Name

Out Params:

Value :Field Value

AddRow() as Long

Purpose: Adds a New Row at end and moves cursor to that row

Return Value:

1-Success
0-Failure

SetRow(RowNo as Long) as Long

Purpose: Moves the cursor to a specified row.

InPut Params:

RowNo: Row no to which cursor to be moved

Return Value:

1-Success
0- Failure (Row does not exists)

Close()

Purpose: Closes the Opened Document.

ApplyFormula() as Long;

Purpose: Calculates , Stores Values of all Calculable fields in current row , using the formulae given in Voucher Wizard.

ApplyFormulaToAllRows() as Long

Purpose: Calculates , Stores Values of all Calculable fields in All Rows, using the formulae given in Voucher Wizard.

Sort(SortDesc as String) as integer

Purpose: Sorts Rows in a Document based on SortDesc

Input Params:

SortDesc:

Syntax: *FieldName Order FieldDataType , FieldName Order*

FieldDataType

FieldName: Transaction Field Name Ex: ProductSeqId,

AccountName

(embedded spaces in Field names not allowed)

Order:

Ascending or Descending Key words : asce , desc

FieldDataType

Key words: text, numb, date

Ex: sort(ProductName desc text,xyzAmount asce Numb , MfDate asce Date)

Default Order: asce, Datatype: text

Direct Methods

DeleteDocument(VoucherType String, VoucherNo as String) as long

Purpose: Deletes an Existing Document.

Input Params:

VoucherType : VoucherType is abbreviation defined in focus for a Voucher. For example, sales invoice Voucher Type is "Sal"

VoucherNo : Voucher Number of document to be deleted.

ReturnValue:

1 : Success

0 : Failure

IsVoucherExisting(VoucherType as String, Vno as String)as Boolean

Purpose: To check whether a document is existing or not

Return Value:

1:Existing.

0:Not Existing

GetNextVoucherNo(VoucherType as String) as String

Purpose: To Get Next Voucher No Of Specified Voucher Type

Field List

Field	Description	Remarks	Set	Get
Date	Invoice/Voucher Date		Y	Y
Voucher	VoucherNo		Y	Y
Account SeqId Account Name Account Code Account Alias			Y	Y
Book SeqId Book Name Book Code Book Alias			Y	Y
Tag SeqId Tag Name Tag Code Tag Alias		Tags: Ex: Department, Cost Centre....	Y	Y
Product SeqId Product Name Product Code Product Alias			Y	Y
Quantity			Y	Y
ProductUnitId			Y	Y
Rate			Y	Y

Amount			Y	Y
DrCr		Dr/Cr Set this Field after setting Amount field only	Y	Y
Batch			Y	Y
Expiry			Y	Y
MfDate			Y	Y
Currency Id , Currency Symbol			Y	Y
Updateinventory		ON/OFF	Y	Y
BRS Status		0/1 (0 means Not - 1 Means Done)	Y	Y
Locked		0/1	Y	Y
Checked		0/1	Y	Y
LinkVoucherNo			N	Y
Reference#		#Ranges from 0 to 255	Y	Y
RefDep#		#Ranges from 0 to 255	Y	Y
RefDueDate#		#Ranges from 0 to 255	Y	Y
RefAmount#		#Ranges from 0 to 255	Y	Y
RefType#	1 -adjust,2-new, 3-onaccount	#Ranges from 0 to 255	Y	Y
RefCount			N	Y
RefHandle			N	Y
AdjAmount		Amount to be adjusted	Y	Y
LinkId			Y	Y
LinkDate			Y	Y

ExtraFields can be accessed through their names (Case Sensitive)

Sample Code

```

Public trans As New Transaction
trans.SetField "Date", Format(Dt_Doc.Value, "dd-MM-yyyy")
trans.SetField "Book Name", optCustAcct.SelName

seqid = fm.OpenMaster("Product", "Name", Trim(.TextMatrix(i, 1)))
If seqid = 0 Then

    fm.New "Job"
    fm.SetField "Name", Trim(.TextMatrix(i, 1))
    'fm.SetField "Code", VarJobId
    fm.Save

End If

trans.SetField "Product Name", "abc"
trans.SetField "Product Code", "a01"
trans.SetField "Material", .TextMatrix(i, 14)
trans.SetField "Unit Labour", .TextMatrix(i, 6)
trans.SetField "Quantity", "120"
trans.SetField "Rate", ".10"

```

```

trans.SetField "Gross", "1200"
trans.SetField "Disc Labor", CDec(txtDiscLabor.Text)
trans.SetField "Disc Material", CDec(txtDiscMat.Text)
trans.AddRow
End If

End With
Next i

flg = trans.SaveDocument
trans.Close

```

FReport

Methods

OpenAccountTransactions(StartDate as Date, EndDate as Date, SeqId as Long)as Long:

Purpose: Opens all Transactions done on an account during the given period. Opens as ReadOnly , Forward Only.

Input Params:

StartDate : Starting Date of period

Enddate : Ending Date of period

SeqId : Account SeqId for which transactions to be retrieved

Return Value:

1-Success

0-No records Found.

OpenProductTransactions(StartDate as Date, EndDate as Date, SeqId as Long)as Long

Purpose: Opens all Transactions done on a product during the given period. Opens as ReadOnly , Forward Only.

Input Params:

StartDate : Starting Date of period

Enddate : Ending Date of period

SeqId : Product SeqId for which transactions to be retrieved

Return Value:

1-Success

0-No records Found.

OpenDocuementList(StartDate as Date, EndDate as Date, VoucherType as String) as Long

Purpose: Opens all Transactions of a voucher type during the given period. Opens as ReadOnly , Forward Only.

Input Params:

StartDate : Starting Date of period
Enddate : Ending Date of period
VoucherType: Voucher type for which transactions to be retrieved

Return Value:

1-Success
0-No records Found.

OpenAllTransactions(StartDate, EndDate) as Long

Purpose: Opens all Transactions during the given period. Opens as
ReadOnly , Forward Only.

Input Params:

StartDate : Starting Date of period
Enddate : Ending Date of period
VoucherType: Voucher type for which transactions to be retrieved

Return Value:

1-Success
0-No records Found.

NextRecord() as Long

Purpose: Moves Cursor to Next Record

Return Value:

1-Success
0-Failure(No More Records)

GetField(Name as String, Value as String) Long

Purpose: To Get value of a Field

Input Params:

Name: Field Name

Out Params:

Value :Field Value

Direct Methods

GetBalanceOn(AcctSeqId as Long, ondate as Date) as double:

Purpose: To get Balance of an Account on a Date

Input Params:

AccSeqId : Account Sequence Id
OnDate : Date

Return Value:

Balance : Balance of Account as on given date.

GetOpeningBalanceOn(AccSeqId as Long, OnDate as Date) as double

Purpose: To get Opening Balance of an Account on a Date

Input Params:

AccSeqId : Account Sequence Id

OnDate : Date

Return Value:

Opening Balance

GetStockOn(ProdSeqId as Long, OnDate as Date , WareHouseSeqId as Long, Quantity as Double, Value as Double):

Purpose: To Get Stock (in terms of Quantity , Value)of a product on a Date, in an Ware House

Input Params:

ProdSeqId :

OnDate :

WareHouseSeqId : 0 – All WareHouses

Output Params:

Quantity: Stock in Base Units of specified Product

Value: Stock in Base Currency

GetReservedStockOn(ProductSeqId as Long , OnDate as Date , WareHouseSeqId as Long, Quantity as Double);

Purpose: To Get Reserved Stock (in terms of Quantity , Value)of a product on a Date, in an Ware House

Input Params:

ProdSeqId :

OnDate :

WareHouseSeqId : 0 – All WareHouses

Output Params:

Quantity: Stock in Base Units of a specified Product

Value: Stock in Base Currency

Fields

Field	Description	Remarks
Date		
VoucherType		
Voucher		
Account SeqId		

Account Name Account Code Account Alias		
Book SeqId Book Name Book Code Book Alias		
Tag SeqId Tag Name Tag Code Tag Alias		
Product SeqId Product Name Product Code Product Alias		
Amount		
DueDate		
Quantity		
Rate		
Batch		
Expiry		
MfDate		
Currency Id, Currency Symbol		
BRS Status		
Locked		
Checked		
NoOfAdjInvs		
AdjInvType		
AdjInvNo		
AdjInvAmount		
AdjInvDate		

Extra Fields can be accessed through their names (Case Sensitive)

Sample Code

```
Dim Fr As New FReport
Dim PQty As Double
Dim AvgVal As Double
Dim varAvgRate As Double
```

\\can be used in a loop

```
fr.GetField "Account Seqid"
fr.NextRecord
```

```
varTotalRecords = fr.OpenAccountTransactions (01-01-2008, 12-12-2007,
varAccoutSeqid)
```

\\Open a doc for specific voucher type

```
VarTotalRecords = fr.OpenDocumentList(Cdate("1-1-2008", cdate("12-12-2007")),
"Sal")
varTotalBal= fr.GetBalanceOn(varAccoutseqid, (30-12-2008))
```

```
Fr.GetStockOn (optProduct.CurSelId, Format(Date, "DD-MM-YYYY"),
               CLng(varBal), PQty, AvgVal)
```

```
Fr.GetReservedStockOn optProduct.CurSelId, Format(Date, "DD-MM-
YYYY"), CLng(varBal), PQty
```

FMasterReport

This class is used to take report of Masters.

Methods

Open(MasterType as String, MasterFlag as Integer, GroupSeqId as Long) as Long

Purpose: Opens Report

InputParams:

MasterType: Master Name Ex: Account , Product, Tag Names

MasterFlag: 1-Groups Only, 2-Leaves Only, 0-All

GroupSeqId: 0 – All groups, Nonzero - specified group Masters will be reported

Return Value:

1-Success

0-Failure

GetField(FieldName as String, FieldValue as String) as Long

Purpose: To Get value of a Field

Input Params:

FieldName

Out Params:

Field Value

MoveNext() Long

Purpose: Moves Cursor to next record

Return Value:

1: Success

0: Failure(No More Records)

Close()

Purpose: Closes the Current Report.

Field List

Field	Description	Remarks
-------	-------------	---------

SeqId		
Name		
Code		
Alias		
Parent SeqId		
Parent Name		
Parent Code		
Parent Alias		
Limit		
limit2		
CreditDays		
Type		
IsGroup		
Level		

ExtraFields can be accessed through their names (Case Sensitive)

Sample Code

```
fmr.Open "Product", 0, 0

While fmr.MoveNext <> 0
  ReDim Preserve stracc(i)
  fmr.GetField "type", temp
  fmr.GetField "Code", temp1
  If (temp = 6 Or temp = 7 Or temp = 5) Then
    fmr.GetField "Name", temp
    stracc(i).accName = temp
    fmr.GetField "SeqId", temp
    stracc(i).accSeqId = temp
    fmr.GetField "Level", temp
    stracc(i).Level = temp
    If stracc(i).Level = 0 Then
      stracc(i).RowPos = i
    Else
      stracc(i).RowPos = stracc(i - 1).RowPos
    End If
    fmr.GetField "IsGroup", temp
    If temp = False Then
      stracc(i).isgrp = 0
    Else
      stracc(i).isgrp = 1
    End If
    fmr.GetField "Parent SeqId", temp
    stracc(i).parent = Val(temp)
  End If
Wend
```

FLinks

Methods

Open(LinkNo as Integer, PartySeqId as Long, ProductSeqId as long, OpenType as Integer) as Long

Purpose: Opens linked/unlinked documents of a given customer/vendor and product where Link Number is number and voucher type is not known or if you wish to open all irrespective of voucher type.

InputParams:

LinkNo : LinkNo in Preferences->Link Definitions
PartySeqId :
ProductSeqId :
OpenType : 2-Pending

Return Value:

Number Of Rows.

Open2(VoucherType as String, PartySeqId as Long, ProductSeqId as Long, OpenType as Integer , LinkSeq as Integer)as Long

Purpose: Opens linked/unlinked documents of a given customer/vendor and product for a given voucher Type.

InputParams:

VoucherType : Link Document Abbreviation
PartySeqId :
ProductSeqId :
OpenType : 2-pending
LinkSeq : 0 – in Single Linking

In Multiple links LinkSeq indicates the order of Link.(Zero based)

Return Value:

Number Of Rows

Open3(VoucherType as String, PartySeqId as Long, ProductSeqId as Long, OpenType Integer, LinkSeq as Integer, Tag1SeqId as Long, Tag2SeqId as Long, Tag3SeqId as long, Tag4SeqId as Long, Tag5SeqId as Long, Tag6SeqId as Long, Tag7SeqId as Long, Tag8SeqId as Long) as Long

Purpose: Opens linked/unlinked documents of a given customer/vendor and product with all the tags that are used in the link definition.

InputParams:

VoucherType : Link Document Abbreviation
PartySeqId :
ProductSeqId :
OpenType : 2-pending
LinkSeq : 0 – in Single Linking
In Multiple links ; LinkSeq indicates the Sequence of Link.(Zero based)

Return Value:

Number Of Rows

Close()

SetRow(nRow as Long)as Long

Purpose: Moves the cursor to a specified row.

Input Params:

nRow: Row no to which cursor to be moved

Return Value:

1-Success

0- Failure (Row does not exists)

GetField(FieldName as String, FieldValue as String) as Long

Purpose: To Get value of a Field

Input Params:

Field Name

Out Params:

Field Value

Field List

FieldName	Description	Remarks
LinkDataRecNo		
LinkValue		
LinkId		
LinkDate		

Sample Code

```
Dim TrRec As New TransRec
Dim fl As New FLinks

For m = 1 To rst.RecordCount
    TotRows = fl.Open3("Sal", rstAcct!Bookno, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    If TotRows = 0 Then
    Else
        '***** start process *****
        For i = 0 To TotRows - 1

            fl.SetRow i
            fl.GetField "linkDataRecNo", LinkRecNo
            TrRec.LoadTransaction (CLng(LinkRecNo))
            TrRec.GetField "Voucher", strData 'broker seqid
            TrRec.GetField "Date", strData
            varVDate = ConvertDate2FocusDate(strData)
        next i
    fl.Close
```

References

Methods

LoadPendingReferences(Code as Long, Tag as Long)as Long

InputParams:

Code : Party Sequence Id
Tag : if using Department wise bill wise accounting
Then pass Dept seqId else zero

Return Value:

Number Of Pending References (No of Rows)

Open(Code as Long, Tag as Long, OpenType as integer) as Long

Purpose: This is used if you have account code and it will return the pending reference numbers of all the documents entered for the account.

InputParams:

Code : Party Sequence Id
Tag : if using Department wise bill wise accounting
Then pass Dept seqId else zero
OpenType: 1-All, 2-Pending

Return Value:

Number Of References (No of Rows)

Open2(AccountSeqId as Long, Tag1SeqId as Long, OpenType as Long, OldRefHandle as Long, OldNoRefs as Integer, ReferenceType as Integer)as Long

Purpose: This is opens the list of references as per given reference numbers and reference types. Using this one can get new references of a given account along with reference numbers.

InputParams :

AccountSeqId : Party Sequence Id
Tag1SeqId : if using Department wise bill wise accounting
Then pass Dept : seqId else zero
OpenType : 1-All, 2-Pending
OldRefHandle : To Edit a transaction's Billwise Details Get "RefHandle" from Transaction class and pass it to this method to get Already adjusted References also, otherwise pass zero
OldNoRefs : To Edit a transaction's Billwise Details Get "RefCount" from Transaction Class and pass it to this method.
Other wise pass zero
ReferenceType : 1- debit(negative) , 2 Credit(positive)

Return Value:

Number Of References (No of Rows)

SetRow(long Row) as Long

Close()

Field List

FieldName	Description	Remarks
Reference Id. (Transaction "Reference")	DataType : Long	
Ref	DataType : String	
VoucherType	DataType: String	
VoucherNo	DataType: String	
DueDate	DataType: String	
Amount	DataType: Double	
Date	DataType: String	
AdjAmount	DataType: Double	
DateRecNo	DataType: Long	

Sample Code

```
Dim RefIR As New References
Dim RefP As Long
Dim RPIB As String

RefP = RefIR.LoadPendingReferences(Tsave(a).CreditAccount, RPIB)
For i = 0 To RefP - 1
    RefIR.SetRow (i)
    strData = RefIR.Ref
    strData = RefIR.Details
    strData = -(RefIR.AdjAmount)
    Ii = Ii + 1
Next i
RefIR.Close
```

Example II

```
Dim FRef As New References
Dim trec As New TransRec
Dim Trans As New Transaction
Dim Fmr As New FMasterReport

Company1.InitializeSelectedCompany

Dim RetVal As Long
Dim VType As String
Dim strVNo As String
Dim nRowsSales As Long
Dim DueDate As String
Dim RefId As String
Dim RefAmount As String
Dim Reference As String
Dim RefType As String
Dim AdjAmount As String
Dim Seqid As String
```

```

Dim i As Long
Dim j As Single
j = 0

Fmr.GetField "Seqid", Seqid
RetVal = FRef.Open2(6, 0, 1, 0, 0, 0) 'CLng(Seqid)
For i = 0 To RetVal - 1
    FRef.SetRow i
    trec.LoadTransaction FRef.DataRecNo

    trec.GetField "vouchertype", VType
    trec.GetField "voucher", strVNo
        'trec.GetField "Reference", Reference
    Trans.LoadDocument VType, strVNo
    Trans.GetField "DueDate", DueDate
    Trans.Close
    trec.GetField "DueDate", DueDate
    j = j + 1
Next i
Fmr.Close
MsgBox "due date set for " & j & "Records"

```

FUnits

Methods

Open(ProductSeqId as Long)as Long
Purpose: Opens units of a product
InputParams:
 ProductSeqId:

Return Value:
 1-Success
 0-Failure

Close()

Save() as Long
Return Value:
 1-Success
 0-Failure

Field List

FieldName	Description	Remarks
UnitName (UnitId as Integer)	DataType : String	Input Param: UnitId, 0= Base

Conversion	DataType : Double	Input Param: unitid as integer
------------	-------------------	--------------------------------

Sample Code

```

Dim varUnit As String
Dim strUnitName As String
Dim varConvFactor As String
Dim FUnt As FUnits
Dim varConv As Double

strUnitName = FUnt.UnitName(0)
FUnt.Open (varProdMID)
strUnitName = FUnt.UnitName(1)
varConv = FUnt.Conversion(1)

```

FCurrencies

Methods

New(OnDate as Date)as Integer

InputParams:

OnDate:

Open(OnDate as Date)as Integer

InputParams :

OnDate:

Close()

Save() as Integer

Return Value:

Properties:

CurrencyName(CurrencyId as Integer)

InputParams:

CurrencyId : Index

Conversion(Index as Integer)

DirectMethods:

GetExchangeRate(CurrencyId as Integer, onDate as Date)as Double

Purpose: To Get Exchange Rate of a currency in Native currency on a given Date.

InputParams:

CurrencyId : Index

OnDate :

GetExchangeRateBySymbol(CurrencySymbol as String , DATE OnDate)as Double

Purpose: To Get Exchange Rate of a currency in Native currency on a given Date with Currency Symbol.

InputParams:

CurrencySymbol :
OnDate :

Sample Code

```
Dim varConRate As Double
Dim fc As New FCurrencies
Dim strDate As String
Dim strAmt As String
Dim varNetAmt As Double
Dim strPost As String
Dim varConStdRate As Double
Dim varOrgAmt As Double

Rcount = trans.LoadDocument(VoucherType, VoucherNo)
If rcount > 0 Then
    For I = 0 To rcount - 1

        trans.SetRow I
        trans.GetField "Amount", strAmt
        trans.GetField "Currency Id", strCurrid
        trans.GetField "Date", strDate

        varConRate = fc.GetExchangeRate(CInt(strCurrid),
                                         CDate(strDate))
        trans.GetField "ExchangeRate", strData
        varConRate = CDb1(strData)

        ''' Conv amt ''' sel curr
        varOrgAmt = CDb1(strAmt) / varConRate
        varNetamtOrg = varNetamtOrg + varOrgAmt
        varNetamtOrg = Abs(varNetamtOrg)

        ''' base curr ''' in dhms
        varConStdRate = fc.GetExchangeRate(CInt(strCurrid),
                                           CDate("01-01-2007"))

        If varConStdRate <> varConRate Then
            strPost = "1"
        Else
            strPost = "0"
        End If

    Next I
End If
trans.Close
```


Sample II

```
Dim varConverRate As Long
Dim varExhRate As Long

varConverRate = fc.Conversion(1)
varExhRate = fc.GetExchangeRate(1, Date)

currency name

strSql = "SELECT DISTINCT Currency From dbo.Curnc"
If rst.State = 1 Then rst.Close
rst.Open strSql, conn, adOpenKeyset, adLockOptimistic
If rst.RecordCount > 0 Then
    For i = 1 To rst.RecordCount
        CurId = rst!Currency
        curname = fc.CurrencyName(CurId)
        rst.MoveNext
    Next i
End If
rst.Close
```

FRateMaster

Methods

New(RateType as Integer, PartySeqId as Long, ProductSeqId as Long, OnDate as Date) as Long

InputParams:

RateType : 0-Buying , 1-Selling
PartySeqId :
ProductSeqId :
OnDate:

Return Value:

1-Success
0-Failure

Open(RateType as Integer, PartySeqId as Long,ProductSeqId As Long, OnDate as Date) as Long

InputParams:

RateType : 0-Buying , 1-Selling
PartySeqId :
ProductSeqId :
OnDate:

Return Value:

1-Success
0-Failure

GetRate(Index as Integer) as double

InputParams:

Index: 0-7
0: Overall Rate
1-7: Value A – Value G

Return value:

Rate.

SetRate(Index as Integer, rate as double)as Long

InputParams:

Index: 0-7
0 : Overall Rate
1-7 : Value A – Value G

Save()as Long

Return Value:

1-Success
0-Failure

Close()

Direct Methods

GetBuyingRate(PartySeqId as Long, ProductSeqId as Long, OnDate as Date) as double

Purpose: To Get Party wise Buying rate on a Date

GetSellingRate(PartySeqId as Long, ProductSeqId as Long, OnDate as Date) as double;

Purpose: To Get Party wise Selling rate on a Date

GetSellingVal(PartySeqId as Long, ProductSeqId as Long, OnDate as Date, Index as Integer) as double

Purpose: To Get Party wise Selling Value on a Date
Index: 1-7 ie., Value A - G

GetBuyingVal(PartySeqId as Long, ProductSeqId as Long, OnDate as Date, Index as Integer)as double

Purpose: To Get Party wise Buying Value on a Date
Index: 1-7 ie., Value A - G

Sample Code

```
Dim retId As Long
```

```

Dim dtVal As Date
Dim j As Integer
Dim tempstr
Dim strRate as string
Dim fmRate As New FRateMaster

itmSqlId = txtHide.Text

dtVal = CDate(txtDtval.Text)
tempstr = Year(dtVal)
tempstr = "01-01-" & tempstr    ' pick for 1st month
dtVal = CDate(tempstr)

retId = fmRate.Open(1, 0, itmSqlId, dtVal)

For j = 0 To 7
    strRate = Format(fmRate.GetRate(j), "0.000")
Next j

Dim varBuyRate As Long

varBuyRate = fmRate.GetBuyingVal(12, varitid, Date, 0)

```

FFixedAssets

Fixed Assets is a Master(Tag).

Create Fixed Assets using FMaster Class then use this Class.

Methods

Open(SeqId as Long)as Long

InputParam:

SeqId: SeqId of Fixed Asset.

Close()

GetField(FieldName as String,FieldValue as String)as Long

SetField(FieldName as String,FieldValue as String)as Long

Save() as String

Field List

Field Name	Description	Remarks	Set	Get
Name			Y	Y
Code			Y	Y
Alias			Y	Y
PurchaseDate			Y	Y
AssetType			Y	Y
Location			Y	Y
Vendor			Y	Y
Qty			Y	Y
QtyLimit			Y	Y
Price			Y	Y
InsuredValue			Y	Y
ScrapValue			Y	Y
PurchaseLimit			Y	Y
JpgPath			Y	Y
AccruedDepreciation			Y	Y
ValueAdded			Y	Y
ValueDeducted			Y	Y
DepreciationMethod			Y	Y
UnitOfUsage			Y	Y
RateOfDepre			Y	Y
Commencement			Y	Y
StartingUnit			Y	Y
AssetAcCode			Y	Y
DepreCode			Y	Y
DeprePICode			Y	Y
SalesAccount			Y	Y
Disposed			Y	Y

Sample Code

```

Public fm As New FMaster
Public FxdAss As New FFixedAssets

count = count
NewCode = VoucherNocount
MRet = fm.OpenMaster("Fixed Assets", "Name", NewCode)
If MRet = 0 Then
    fm.New "Fixed Assets"
    fm.SetField "Name", NewCode
    fm.SetField "Code", NewCode
    fm.SetField "Type", "1" ' For asset type= FA
    fm.SetField "Purch. Date", CStr(VDate) ' purchase dt
    fm.SetField "Vendor", VendorName ' set vendor SeqId on Fassets screen
    fm.SetField "AccruedDepreciation", "0" 'Rate ' Cum depr

```

```

        fm.SetField "Price", Rate
        fm.SetField "Qty", "1"
        fm.SetField "AssetAcCode", AccName ' purchas acc no
        fm.SetField "DepreCode", "104"
        fm.SetField "DeprePICode", "1818"
        fm.SetField "Status", "N"
        fm.SetField "DepreciationMethod", "3" 'stragiht lin
        fm.SetField "RateOfDepre", 3 ' write of in yrs
        fm.SetField "Location", varLoc
        fm.SetField "Fst Description", strfstDesc
        fm.SetField "Model Number", ProdseqID ' to set model no with prod
name
        Seqid = fm.Save
        fm.Close

        If Seqid > 0 Then '***** No need to update extra fields here gets change
when fm opens

        FxdAss.Open Seqid
        FxdAss.SetField "Type", "1"
        FxdAss.SetField "Vendor", VendorName
        FxdAss.SetField "Purch. Data", CStr(Format(VDate, "dd-MM-yyyy"))
        FxdAss.SetField "Price", Rate
        FxdAss.SetField "AccruedDepreciation", "0" 'Rate
        FxdAss.SetField "Qty", "1"
        FxdAss.SetField "AssetAcCode", AccName
        FxdAss.SetField "DepreCode", "104" 'AccName ' 'Accumulated
Depreciation - Dubai
        FxdAss.SetField "DeprePICode", "1818" 'D'epreciation Expense -
Dubai
        FxdAss.SetField "DepreciationMethod", "3"
        FxdAss.SetField "RateOfDepre", 3
        FxdAss.SetField "Location", varLoc
        FxdAss.Save
        FxdAss.Close
    End If

```

FPayroll

Methods

New(EmpSeqId as Long, OnDate as DATE) as Long

Input Params:

EmpSeqId:

OnDate : Payroll Date.

Return Value:

0-Payroll for this employee is existing for this period

1-Success.

Open(EmpSeqId as Long , OnDate as DATE) as Long

Input Params:

EmpSeqId:

OnDate: Payroll Date.

Return Value:

1-Success

0- Payroll for this employee is not existing for this period

Save() as Long

Return Value:

1 - Success

0 - Failure

Field List

Name	DataType	Description	Remarks	Set	Get
NoOfWorkingDays	Single			N	Y
NoOfDaysWorked	Single			y	Y
OTHrs	Single			y	y
HOTHrs	Single				
Basic	Double			n	y
EarnedBasic	Double			n	y
OTAmount	Double			n	y
HOTAmount	Double			n	y
NetAmount	Double			n	y
Remarks	String			y	y
MonthlyInputs	Double			y	y
DailyInputs	Double			y	y
Leaves	Single			y	y
Loans	Double			y	y
Earnings	Double			n	y
Deductions	Double			n	y

Sample

Basic details

```
Dim femp As FPayroll  
Dim strempData As String
```

```
femp.Open (1)
```

```

strempData =femp.Basic
strempData =femp.DailyInputs (1)
strempData =femp.HOTAmount
strempData =femp.HOTHrs
strempData =femp.NoOfDaysWorked
strempData =femp.OTHrs

femp.Close

```

Sample II

```

Trans.GetField "Employees Seqid", empid
emp.Open CLng(empid)
Salary = emp.Basic
emp.Close
Trans.SetField "Rate1", CStr(Salary)

```

Sample III

Getting emp name with their categories

```

Dim CatCode As String
Dim CatName As String
Dim Cnt As Integer
Dim StrAA As String
Dim k As Integer
Dim J As Integer

Cnt = 0
Dim mis As New MiscellaneousMod
mis.GetCategories CatCode, CatName
Cnt = UBound(Split(CatName, ",")) + 1

k = 1
J = 0
'''name into array
For i = 1 To Len(CatName)
    If Mid(CatName, i, 1) = "," Then
        StrAA = Mid(CatName, k, InStr(i, CatName, ",") - k)
        k = i + 1
        ReDim Preserve Nam(J)
        Nam(J) = StrAA
        J = J + 1
    End If
Next i
StrAA = Mid(CatName, k, Len(CatName) - k + 1)
ReDim Preserve Nam(J)
Nam(J) = StrAA
'''Code into array
k = 1
J = 0
For i = 1 To Len(CatCode)
    If Mid(CatCode, i, 1) = "," Then
        StrAA = Mid(CatCode, k, InStr(i, CatCode, ",") - k)
        k = i + 1
        ReDim Preserve Code(J)

```

```

        Code(J) = StrAA
        J = J + 1
    End If

Next i
StrAA = Mid(CatCode, k, Len(CatCode) - k + 1)
ReDim Preserve Code(J)
Code(J) = StrAA
For i = 0 To Cnt - 1
    If Nam(i) = "Technician" Then
        EmpCode = Code(i)
    End If
    If Nam(i) = "Operation" Then
        EmpCodeOper = Code(i)
    End If
Next i

```

LOGINS

To get user user/login details

Methods

Properties	Explanation	Parameter/ Return Type
GetCurrentUser	To Get current login user id	Int
GetCurrentUserName	To Get current login user name	String
Open	Open list for all the users available in focus	Long
ReadNext	To get next user record after initializing login class method	Boolean
GetId	To Get user id after opening login class	Int
GetName	To Get user name after opening login class	String
GetParent	To Get Parent Id	Integer
IsGroup	User associated to any grp or mot	Boolean
LoadUser	Open specific user object	String
GetVoucherPrefix	Gives Voucher details assigned to that user	String /Buffer

Sample Codes

Sub Loadusers()

```

Dim fu As New Logins
Dim stru As String
Dim rowCnt As Integer

```



```

Dim strUserchk As String

I = 1
rowCnt = 1

Seq = fu.Open
fu.ReadNext
grd.Rows = Seq + 1
With grd

    Do While I <> Seq

        uid = fu.GetId
        fu.GetName stru

        If fu.IsGroup = False Then

            .TextMatrix(rowCnt, 0) = rowCnt
            .TextMatrix(rowCnt, 1) = uid
            .TextMatrix(rowCnt, 2) = stru

        fu.ReadNext
        I = I + 1

    Loop

End With

End Sub

```

FRemoteExec

Class available in Focus Api to connect without a DSN and database password

Methods

Properties	Explanation	Parameter/ Return Type
AddColumn	<p>To identify column name and column type .</p> <p>Fst param = Data type , which focus Api needs (If column type is varchar or text =1, 8 numeric, 4 int , 91 date)</p> <p>Sec param = Data length of column in database table</p> <p>Third param = Column Name</p>	rc.AddColumn 4 , 8, "Column NAME"
Execute	To execute sql stmt	String
GetValue	To get data from column	rc.GetValue "Column name", strName
FetchNext	To get loop moving cursor	
Reset	Need to call before update	0

	stmt	
--	------	--

Sample Code

```
Dim rc As New FRemoteExec
```

```
Public Function ChkData(ByVal hwnd As Long, ByVal Product As Long, ByVal SelDate As Long) As Long
```

```
Dim strName As String
Dim retid As Long
```

```
rc.Reset (3) ' total no of column u want to read or update
```

```
rc.AddColumn 4, 8, "No_Agents" '***** No_Agent = Column name
                               '***** 4 = Column Type is int (if column type is
                               'varchar or text , 1, ' 1 = varchar , 8 numeric, 4 int , 91 date)
                               '*****
```

```
rc.AddColumn 4, 8, "No_AgentsID"
rc.AddColumn 91, 0, "DT_Agent"
```

```
retid = rc.Execute("Select No_Agents,No_AgentsID,DT_Agent from tbl_Agents where
AcctID = 68")
While rc.FetchNext()
```

```
    rc.GetValue "No_Agents", strName
    rc.GetValue "No_AgentsID", strName
    rc.GetValue "DT_Agent", strName
```

```
Wend
rc.Close
```

```
rc.Reset (0) ' Reset b4 giv any updation stmt
```

```
rc.Execute "update tbl_Agents set No_Agents = 40 , No_AgentsID =4 where AcctID = 68
"
```

```
rc.Execute "delete from tbl_Agents where AcctID = 79 "
```

```
rc.Execute "insert into tbl_Agents values (12,69,76, "" & CDate("12-12-2005") & "") "
```

```
End Function
```

