

JUST-IN-TIME SOLUTIONS

COREAPI SPECIFICATION DATABASE DOCUMENT

Data Dictionary

29/04/2021

TRIAL

TRIAL

TRIAL

Table of contents

FAST@180.77.15.102	9
1. User	10
1.1. Tables	11
1.1.1. Table: C_COUNTRY (Country)	11
1.1.2. Table: S_BRANCH (Branch of the bank)	12
1.1.3. Table: S_DEPARTMENT	14
1.1.4. Table: S_USERPOLICY (Policy of the user)	15
1.1.5. Table: S_USERROLE (User Role)	17
1.1.6. Table: S_USRAC (User Profile)	18
1.1.7. Table: S_USRSSN (User session)	20
1.2. Procedures	21
1.2.1. Procedure: ADM_GET_ISLOGIN (Get Is Login)	21
1.2.2. Procedure: ADM_USER_DELETE (Delete User Profile)	22
1.2.3. Procedure: ADM_USER_INSERT (Create new user profile)	23
1.2.4. Procedure: ADM_USER_LOGIN (User Login to System)	26
1.2.5. Procedure: ADM_USER_UPDATE (Update User Profile)	27
1.2.6. Procedure: ADM_USERS_SEARCH (Search User Profile)	29
2. IPC	30
2.1. Tables	30
2.1.1. Table: IPC_SCHEDULEACTION (Schedule Action)	30
2.1.2. Table: IPC_SCHEDULEDETAIL (Schedule Detail)	31
2.1.3. Table: IPC_SCHEDULEDETAILHIS (Schedule detail history)	32
2.1.4. Table: IPC_SCHEDULEREPEAT (Schedule Repeat)	33
2.1.5. Table: IPC_SCHEDULES (Schedule Information)	34
2.1.6. Table: IPC_SCHEDULESHIS (Schedule History)	35
2.1.7. Table: IPC_SWIFT_LOGDEFINE (IPC Swift Log Define)	36
2.1.8. Table: IPCCOMPONENT (IPC component)	37
2.1.9. Table: IPCCONNECTIONWS (IPC Connection WebService)	38
2.1.10. Table: IPCLOGSYNCDEFINE (IPC Log Sync Define)	39
2.1.11. Table: IPCOUTPUTDEFINEHTTP (IPC Output Define HTTP)	40
2.1.12. Table: IPCSYNCSEQ (IPC SYNC Sequences)	41
2.1.13. Table: IPCTRANCODEMAPING_ISO (IPC transacion code mapping ISO)	42
2.1.14. Table: SYSERRORLIST (System Error List)	43
2.2. Procedures	44
2.2.1. Procedure: IPC_GET_LISTSCHEDULE (Get List Schedule Information)	44
2.2.2. Procedure: IPC_GETNEXTTIMEEXEC (Get Next Time Execute)	45
2.2.3. Procedure: IPC_UPDATELOGDEFINE_EVENT (IPC Update Log Define Event)	46
2.2.4. Procedure: IPCLOGTRANS_UPDATESTATUS (Log transaction Update Status)	53
2.2.5. Procedure: IPCLOGTRANS_UPDATESTATUS_SYNC TO CORE (Log transaction update status sync to core)	54
3. SYSTEM	55
3.1. Tables	55
3.1.1. Table: S_BANK (Bank Information)	55
3.1.2. Table: S_BRANCH (Branch of the bank)	56
3.1.3. Table: S_PARAM (System parameter)	58
3.1.4. Table: SYSERRORLISTSOURCE (System error list source)	59

3.2. Procedures	60
3.2.1. Procedure: SYS_GET_BANK (Get Bank List)	60
4. FAST	61
4.1. Tables	61
4.1.1. Table: FAST_BANK (FAST Bank)	61
4.2. Procedures	62
4.2.1. Procedure: FAST_BANKLIST (FAST Bank List)	62
4.2.2. Procedure: FAST_GETOUTGOING (FAST GET Outgoing Transaction)	63
4.2.3. Procedure: FAST_GETTRANBYFCNTXREFID (Get Transaction By FCN)	64
4.2.4. Procedure: FAST_GETTRANBYPMTINFID (Get Transaction Payment ID)	65
4.2.5. Procedure: FAST_GETTRANDETAIL (FAST Get transaction detail)	66
4.2.6. Procedure: FAST_SELECTTRANBYPAYMENTID (FAST Select transaction payment ID)	67
4.2.7. Procedure: FAST_SELECTTRANCODEMAPPINGISO (FAST Get transaction code mapping ISO)	68
4.2.8. Procedure: IPCLOGTRANS_GETTRANSACTION_TOFAST (Get transaction send to FAST)	69
5. Other	70
5.1. Tables	70
5.1.1. Table: EBA_ROLES	70
5.1.2. Table: EBA_TRANRIGHT	71
5.1.3. Table: IPC_MAPPING_SWIFT	72
5.1.4. Table: IPC_SCHEDULEACTIONPARA	73
5.1.5. Table: IPC_SCHEDULEDAY	74
5.1.6. Table: IPC_SCHEDULEDAYHIS	75
5.1.7. Table: IPC_SCHEDULEREPEATHIS	76
5.1.8. Table: IPC_TIMEOUTTRANS	77
5.1.9. Table: IPCBLSCHECKREVERSAL	78
5.1.10. Table: IPCBLSREVERSALFLOW	79
5.1.11. Table: IPCCONNECTIONDB	80
5.1.12. Table: IPCDATADEFINE	81
5.1.13. Table: IPCINPUTDEFINEHTTP	82
5.1.14. Table: IPCINPUTDEFINEXML	83
5.1.15. Table: IPCINPUTDEFINEXML_ISO_MAPTRANCODE	84
5.1.16. Table: IPCINVENTORY	85
5.1.17. Table: IPCLOGDEFINE	86
5.1.18. Table: IPCLOGMESSAGE	87
5.1.19. Table: IPCLOGMESSAGEHIS	88
5.1.20. Table: IPCLOGTRANAPPDETAIL	89
5.1.21. Table: IPCLOGTRANAPPDETAILHIS	90
5.1.22. Table: IPCLOGTRANAPPROVE	91
5.1.23. Table: IPCLOGTRANAPPROVEHIS	92
5.1.24. Table: IPCLOGTRANS	93
5.1.25. Table: IPCLOGTRANDETAIL	99
5.1.26. Table: IPCLOGTRANDETAILFIELD	100
5.1.27. Table: IPCLOGTRANDETAILHIS	101
5.1.28. Table: IPCLOGTRANSEVENT	102
5.1.29. Table: IPCLOGTRANSHIS	107
5.1.30. Table: IPCLOGTRANSREVERSAL	112

5.1.31. Table: IPCLOGTRANSREVERSALHIS	113
5.1.32. Table: IPCLOGTRANSREVERSALTELLER	114
5.1.33. Table: IPCMAPPINGCOLNAME	115
5.1.34. Table: IPCOUTPUTDEFINEHAS	116
5.1.35. Table: IPCOUTPUTDEFINEHTTPHEADER	117
5.1.36. Table: IPCOUTPUTDEFINEXML	118
5.1.37. Table: IPCOUTPUTMODIFYHTTP	119
5.1.38. Table: IPCPARMINFO	120
5.1.39. Table: IPCSERVICE	121
5.1.40. Table: IPCSYSVAR	122
5.1.41. Table: IPCTRANSLIST	123
5.1.42. Table: O9\$ERRORCODE	124
5.1.43. Table: S_SYSTEMCODE	125
5.1.44. Table: SMP_MASTERPAGE	126
5.1.45. Table: SMP_PAGES	127
5.1.46. Table: SMP_POSITION	129
5.1.47. Table: SMP_Settings	130
5.1.48. Table: SMP_THEME	131
5.1.49. Table: SMP_USERINROLE	132
5.1.50. Table: SMP_USERS	133
5.1.51. Table: SMP_WIDGETPAGE	135
5.1.52. Table: SMP_WidgetRight	136
5.1.53. Table: SMP_WIDGETS	137
5.1.54. Table: SMP_WIDGETSLANG	138
5.2. Procedures	139
5.2.1. Procedure: FAST_GET_STATUS	139
5.2.2. Procedure: FAST_RESEND_TX	140
5.2.3. Procedure: FAST_SEARCH_SCHEDULE	141
5.2.4. Procedure: FAST_UPDATE_SCHEDULE	142
5.2.5. Procedure: IPC_CHECKAPPROVED_SCHEDULE	143
5.2.6. Procedure: IPC_CHECKDESTTRANS	144
5.2.7. Procedure: IPC_CHECKREVERSAL	146
5.2.8. Procedure: IPC_CHECKTRANAPPROVE	147
5.2.9. Procedure: IPC_GETDESTTRANREF	149
5.2.10. Procedure: IPC_GETLISTREVERSAL	150
5.2.11. Procedure: IPC_GETNEXTEXECSCSCHEDULE	151
5.2.12. Procedure: IPC_GETNEXTTIMEEXEC_bk20171121	153
5.2.13. Procedure: IPC_GETTRANSID_TEMP	154
5.2.14. Procedure: IPC_GETURL_BYSOURCETRANCODE	155
5.2.15. Procedure: IPC_GetWorkingDate	156
5.2.16. Procedure: IPC_RESETID	157
5.2.17. Procedure: IPC_SCHEDULELOADACTION	158
5.2.18. Procedure: IPC_SCHEDULESGETCURRENT	159
5.2.19. Procedure: IPC_SCHEDULESLOGIN	162
5.2.20. Procedure: IPC_TIMEOUTTRANS_SELECT	165
5.2.21. Procedure: IPC_UPDATE_SESSION	166

5.2.22. Procedure: IPC_UPDATEBATCHTRANS	167
5.2.23. Procedure: IPC_UPDATEREVERSAL	168
5.2.24. Procedure: IPC_UPDATELOGDEFINE	170
5.2.25. Procedure: IPC_UPDATELOGTRANSError	177
5.2.26. Procedure: IPCAPPROVETRAN_TELLER	178
5.2.27. Procedure: IPCAPPROVETRAN_USER	180
5.2.28. Procedure: IPCCHECKEXIST_TRANREF	182
5.2.29. Procedure: IPCCOMPONENT_SELECT	183
5.2.30. Procedure: IPCCOMPONENTSYN_SELECT	184
5.2.31. Procedure: IPCGETMAPPINGCOLNAME	185
5.2.32. Procedure: IPCGETSEQVALUE	186
5.2.33. Procedure: IPCGETSEQVALUE_SCALAR	187
5.2.34. Procedure: IPCLOGMESSAGE_INSERT	188
5.2.35. Procedure: IPCLOGTRANAPPDETAIL_INSERT	189
5.2.36. Procedure: IPCLOGTRANS_GETTRANSSYNC09	190
5.2.37. Procedure: IPCLOGTRANS_GETTRANSSYNCREFUND	191
5.2.38. Procedure: IPCLOGTRANS_INSERT	192
5.2.39. Procedure: IPCLOGTRANS_INSERT_EVENT	196
5.2.40. Procedure: IPCLOGTRANS_OFFLINE_SELECT	203
5.2.41. Procedure: IPCLOGTRANS_OFFLINE_UPDATE	204
5.2.42. Procedure: IPCLOGTRANS_REVERSAL	205
5.2.43. Procedure: IPCLOGTRANS_SYNC09_UPDATE	206
5.2.44. Procedure: IPCLOGTRANS_TIMEOUT	207
5.2.45. Procedure: IPCLOGTRANS_UPDATE	208
5.2.46. Procedure: IPCLOGTRANS_UPDATE_EVENT	209
5.2.47. Procedure: IPCLOGTRANS_UPDATE_REVERSE_REFERENCE	210
5.2.48. Procedure: IPCLOGTRANS_UPDATEAPPROVE	211
5.2.49. Procedure: IPCLOGTRANS_UPDATEFATHERTRANS	212
5.2.50. Procedure: IPCLOGTRANS_UPDATETIMEOUT	214
5.2.51. Procedure: IPCLOGTRANSDetail_INSERT	215
5.2.52. Procedure: IPCLOGTRANSDetail_SELECT	218
5.2.53. Procedure: IPCLOGTRANSDetailFIELD_SELECT	219
5.2.54. Procedure: IPCSCHEDULE_UPDATENEXTEXEC	220
5.2.55. Procedure: IPCSYSVAR_INSERT09SESSION	221
5.2.56. Procedure: IPCSYSVAR_SELECT	222
5.2.57. Procedure: IPCSYSVAR_UPDATE	223
5.2.58. Procedure: PAGES_CHECKPERMIT	224
5.2.59. Procedure: Pages_GetPageInfo	225
5.2.60. Procedure: POSITION_LOAD	226
5.2.61. Procedure: Settings_LoadPortalSettings	227
5.2.62. Procedure: USERS_GETROLE	228
5.2.63. Procedure: WIDGETRIGHT_GETMODULEINPAGEFORROLE	229
5.2.64. Procedure: WIDGETS_LOAD	230
5.3. Functions	231
5.3.1. Function: fn_ParseText2Table	231
5.3.2. Function: Split	233

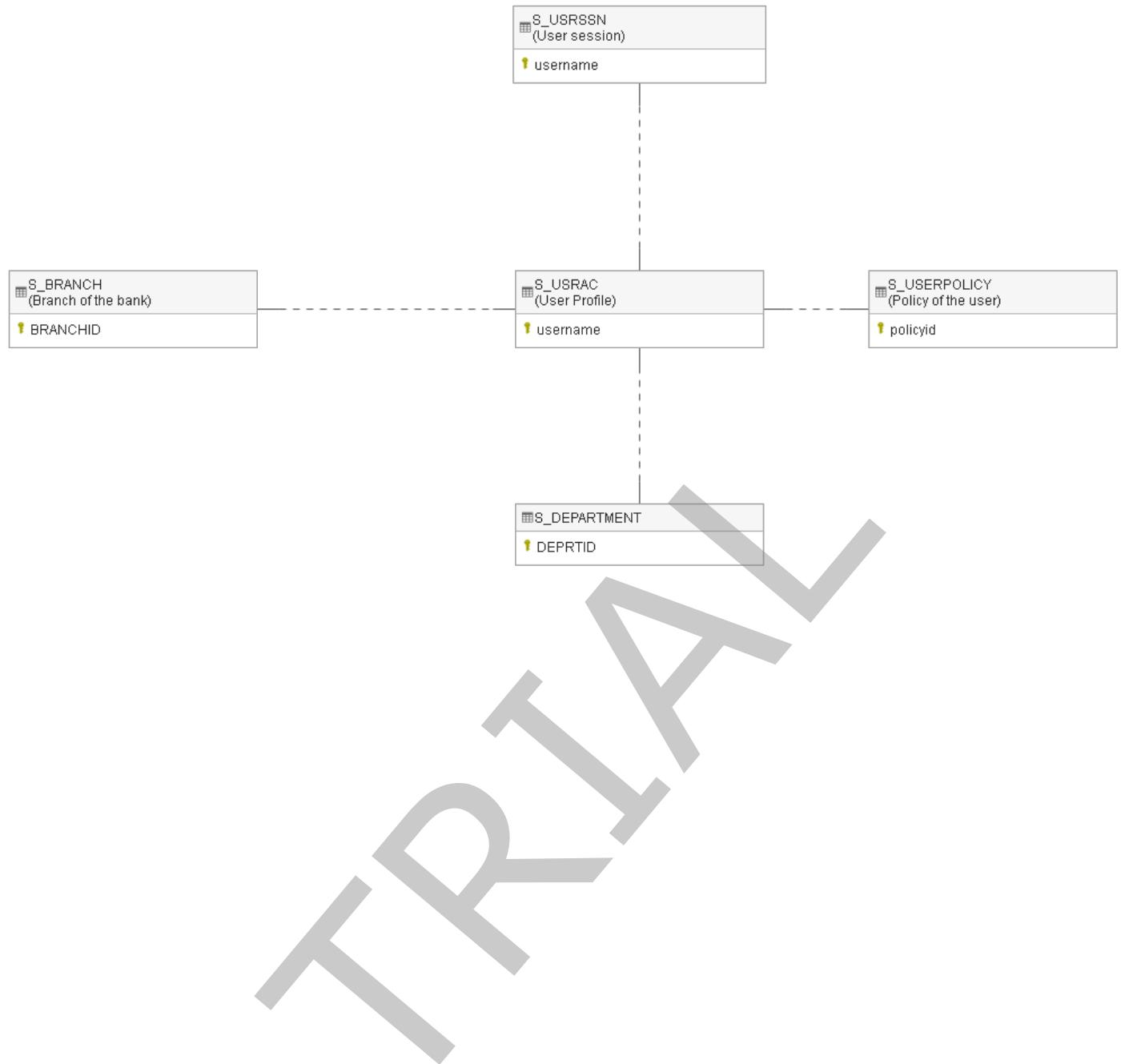
Legend

- 🔑 Primary key
- 🔑 Primary key disabled
- 🔑 User-defined primary key
- 🔑 Unique key
- 🔑 Unique key disabled
- 🔑 User-defined unique key
- ⚡ Active trigger
- ⚡ Disabled trigger
- Many to one relation
- User-defined many to one relation
- ← One to many relation
- User-defined one to many relation
- ↔ Many to many relation
- ↔ User-defined many to many relation
- One to one relation
- User-defined one to one relation
- @ Input
- @ Output
- @ Input/Output
- ↳ Uses dependency
- ↳ User-defined uses dependency
- ↳ Used by dependency
- ↳ User-defined used by dependency

PTIAL

TRIAL

1. User



1.1. Tables

1.1.1. Table: C_COUNTRY (Country)

Sample field: Status: Active

Columns

	Name	Data type	Description / Attributes
█	CTRYCD	nchar(2)	
█	CTRYCD3	nchar(3)	
█	CTRYNAME	nvarchar(20)	
█	MCTRYNAME	nvarchar(MAX)	Nullable
█	CTRSNAME	nvarchar(50)	Nullable
█	MCTRSNAME	nvarchar(MAX)	Nullable
█	CCRID	nchar(3)	
█	LANG	nchar(2)	Nullable
█	CTRYSTS	nchar(1)	
█	ORD	int	Nullable
█	REGION	nchar(1)	Nullable

Unique keys

	Columns	Name / Description
█	CTRYCD	PK_C_COUNTRY

1.1.2. Table: S_BRANCH (Branch of the bank)

Sample field: Status: Active

- This table is show branch of the bank

Columns

	Name	Data type	Description / Attributes
█	BRANCHID	int	
█	BRANCHCD	nvarchar(5)	
█	REFID	nvarchar(15)	Nullable
█	BRNAME	nvarchar(500)	Nullable
█	ADDRESS	nvarchar(500)	Nullable
█	PHONE	nvarchar(255)	Nullable
█	MPHONE	nvarchar(255)	Nullable
█	TAXCODE	nvarchar(50)	Nullable
█	BCYCID	nchar(3)	Nullable
█	BCYNM	nchar(3)	Nullable
█	LCYCID	nchar(3)	Nullable
█	LCYNM	nchar(3)	Nullable
█	REFCODE	nvarchar(250)	Nullable
█	COUNTRY	nchar(2)	Nullable
█	LANG	nchar(2)	Nullable
█	TIMEZN	int	Nullable
█	NUMFMTT	nchar(1)	Nullable
█	NUMFMTD	nchar(1)	Nullable
█	DATEFMT	nvarchar(10)	Nullable
█	LDATEFMT	nvarchar(10)	
█	TIMEFMT	nvarchar(10)	Nullable
█	ISONLINE	nchar(1)	Nullable
█	UDFIELD1	nvarchar(MAX)	Nullable

Linked from

	Table	Join	Title / Name / Description
→	S_USRAC (User Profile)	S_BRANCH BRANCHID = S_USRACbranchid	fk_S_BRANCH_S_USRAC

Unique keys

	Columns	Name / Description
█	BRANCHID	PK_S_BRANCH

Used By

	Name
█	S_BRANCH (Branch of the bank)
⚙	ADM_USER_LOGIN (User Login to System)

TRIAL

1.1.3. Table: S_DEPARTMENT

Columns

Name		Data type	Description / Attributes
DEPRTID	DEPRTID	int	
DEPRTCD	DEPRTCD	nvarchar(5)	
BRANCHID	BRANCHID	int	
DEPRNAME	DEPRNAME	nvarchar(MAX)	Nullable

Linked from

Table	Join	Title / Name / Description
S_USRAC (User Profile)	S_DEPARTMENT DEPRTID = S_USRACdeptid	fk_S_DEPARTMENT_S_USRAC

Unique keys

Columns	Name / Description
DEPRTID	PK_S_DEPARTMENT

Used By

Name
S_DEPARTMENT
ADM_USER_LOGIN (User Login to System)
S_USRAC (User Profile)

1.1.4. Table: S_USERPOLICY (Policy of the user)

Sample field: Status: Active

- This table store User policy

Columns

	Name	Data type	Description / Attributes
PK	policyid	int	
	descr	varchar(200)	Nullable
	isdefault	bit	Nullable
	efffrom	datetime	
	efto	datetime	Nullable
	pwdhis	int	
	pwdagemax	int	
	minpwdlen	int	
	pwdcplx	bit	
	pwdcplxc	bit	
	pwdcplxuc	bit	
	pwdcplxsc	bit	
	pwccplxsn	bit	
	timelginrequire	bit	
	lginfr	varchar(50)	
	lginto	varchar(50)	
	llkoutthrs	varchar(10)	
	resetlkout	varchar(10)	
	usercreate	varchar(50)	Nullable
	datecreate	datetime	Nullable
	usermodify	varchar(50)	Nullable
	datemodify	datetime	Nullable
	contractid	varchar(50)	Nullable
	isbankedit	bit	Nullable
	iscorpedit	bit	Nullable
	baseonpolicy	int	Nullable
	pwdft	nvarchar(100)	Nullable

Linked from

Table	Join	Title / Name / Description
S_USRAC (User Profile)	S_USERPOLICY.policyid = S_USRAC.policyid	fk_S_USERPOLICY_S_USRAC

Unique keys

	Columns	Name / Description
PK	policyid	pk_policy

Used By

Name
 S_USERPOLICY (Policy of the user)
 S_USRAC (User Profile)

TRIAL

1.1.5. Table: S_USERROLE (User Role)

Sample field: Status: Active

- This is filed display role of user

Columns

	Name	Data type	Description / Attributes
█	roleid	int	Identity / Auto increment
█	rolename	nvarchar(1000)	
█	roledescription	nvarchar(4000)	Nullable
█	usertype	varchar(50)	Nullable
█	contractno	varchar(50)	Nullable
█	usercreated	varchar(50)	Nullable
█	datecreated	datetime	Nullable
█	usermodified	varchar(50)	Nullable
█	datemodified	datetime	Nullable
█	serviceid	varchar(50)	Nullable
█	status	varchar(50)	Nullable
█	isshow	varchar(5)	Nullable

Unique keys

	Columns	Name / Description
█	roleid	pk_s_userrole

1.1.6. Table: S_USRAC (User Profile)

Sample field: Status: Active

- This table show all user of system

Columns

	Name	Data type	Description / Attributes
1	userid	int	Identity / Auto increment
2	username	varchar(50)	
3	password	varchar(50)	
4	firstname	nvarchar(1000)	
5	lastname	nvarchar(1000)	Nullable
6	gender	int	Nullable
7	address	nvarchar(4000)	Nullable
8	email	varchar(100)	Nullable
9	birthday	datetime	Nullable
10	phone	varchar(50)	Nullable
11	lastlogintime	datetime	Nullable
12	status	varchar(1)	Nullable
13	usercreated	varchar(50)	Nullable
14	datecreated	datetime	Nullable
15	usermodified	varchar(50)	Nullable
16	datemodified	datetime	Nullable
17	islogin	bit	Nullable
18	expiretime	datetime	Nullable
19	branchid	int	References: S_BRANCH (Branch of the bank)
20	deptid	int	Nullable References: S_DEPARTMENT
21	userlevel	int	Nullable
22	coreuserid	varchar(50)	Nullable
23	corelgnname	varchar(255)	Nullable
24	productid	varchar(50)	Nullable
25	isshow	bit	Nullable
26	policyid	int	Nullable References: S_USERPOLICY (Policy of the user)
27	failnumber	int	Nullable

Links to

	Table	Join	Title / Name / Description
1	S_BRANCH (Branch of the bank)	S_USRAC branchid = S_BRANCHBRANCHID	fk_S_BRANCH_S_USRAC
2	S_DEPARTMENT	S_USRAC deptid = S_DEPARTMENTDEPTID	fk_S_DEPARTMENT_S_USRAC

Table	Join	Title / Name / Description
— S_USERPOLICY (Policy of the user)	S_USRAC policyid = S_USERPOLICYpolicyid	fk_S_USERPOLICY_S_USRAC

Linked from

Table	Join	Title / Name / Description
— S_USRSSN (User session)	S_USRAC Username = S_USRSSNusername	fk_S_USRAC_S_USRSSN

Unique keys

Columns	Name / Description
🔑 username	PK_S_USRAC

Uses

Name
█ S_USRAC (User Profile)
— S_BRANCH (Branch of the bank)
— S_DEPARTMENT
— S_USERPOLICY (Policy of the user)

Used By

Name
█ S_USRAC (User Profile)
⚙ ADM_GET_ISLOGIN (Get Is Login)
⚙ ADM_USER_DELETE (Delete User Profile)
⚙ ADM_USER_INSERT (Create new user profile)
⚙ ADM_USER_LOGIN (User Login to System)
⚙ ADM_USER_UPDATE (Update User Profile)
⚙ ADM_USERS_SEARCH (Search User Profile)
— S_USRSSN (User session)

1.1.7. Table: S_USRSSN (User session)

Sample field: Status: Active

- This table show session of one user

Columns

		Name	Data type	Description / Attributes
	username		nvarchar(50)	References: S_USRAC (User Profile)
	password		nvarchar(200)	
	sessionid		nvarchar(MAX)	Nullable
	busdate		nvarchar(20)	Nullable
	branchcd		nvarchar(10)	Nullable

Links to

Table		Join	Title / Name / Description
	S_USRAC (User Profile)	S_USRSSN username = S_USRACusername	fk_S_USRAC_S_USRSSN

Unique keys

Columns		Name / Description
	username	PK_s_usrssn

Uses

Name	
	S_USRSSN (User session)
	S_USRAC (User Profile)

Used By

Name	
	S_USRSSN (User session)
	IPC_UPDATE_SESSION

1.2. Procedures

1.2.1. Procedure: ADM_GET_ISLOGIN (Get Is Login)

Sample field: Status: Active

- This function used to check one user have login yet

Input/Output

	Name	Data type	Description
@	username	varchar(50)	
@	expiretime	varchar(50)	

Uses

	Name
⚙️	ADM_GET_ISLOGIN (Get Is Login)
👤	s_usrac (User Profile)

Script

```
CREATE PROC [dbo].[ADM_GET_ISLOGIN]
@username VARCHAR(50),
@expiretime VARCHAR(50)
AS
SET DATEFORMAT DMY

SELECT *
FROM s_usrac
WHERE CAST(@expiretime AS DATETIME)>expiretime AND username=@username
```

1.2.2. Procedure: ADM_USER_DELETE (Delete User Profile)

Sample field: Status: Active

- This function used to delete one user profile

Input/Output

Name	Data type	Description
@USERNAME	varchar(50)	

Uses

Name
 ADM_USER_DELETE (Delete User Profile)
 S_USRAC (User Profile)

Script

```
CREATE PROCEDURE [dbo].[ADM_USER_DELETE]
    @USERNAME VARCHAR(50)
AS
SET NOCOUNT ON
DELETE FROM [dbo].[S_USRAC] WHERE [USERNAME] = @USERNAME;
```

1.2.3. Procedure: ADM_USER_INSERT (Create new user profile)

Sample field: Status: Active

- This function used to Insert new user profile to system

Input/Output

	Name	Data type	Description
@	usrid	int	
@	username	varchar(50)	
@	password	varchar(50)	
@	firstname	nvarchar(1000)	
@	lastname	nvarchar(1000)	
@	gender	int	
@	address	nvarchar(4000)	
@	email	varchar(100)	
@	birthday	datetime	
@	phone	varchar(50)	
@	lastlogintime	datetime	
@	status	varchar(1)	
@	usercreated	varchar(50)	
@	datecreated	datetime	
@	usermodified	varchar(50)	
@	datemodified	datetime	
@	islogin	varchar(50)	
@	expiretime	datetime	
@	branchid	varchar(50)	
@	deptid	varchar(50)	
@	userlevel	int	
@	coreuserid	varchar(50)	
@	corelname	varchar(255)	
@	productid	varchar(50)	
@	isshow	varchar(5)	
@	policyid	int	
@	failnumber	int	

Uses

Name
 ADM_USER_INSERT (Create new user profile)
 S_USRAC (User Profile)

Script

```
CREATE PROCEDURE [dbo].[ADM_USER_INSERT]
    @usrid int,
    @username varchar(50),
```

```
@password varchar(50),
@firstname nvarchar(1000),
@lastname nvarchar(1000),
@gender int,
@address nvarchar(4000) ,
@email varchar(100) ,
@birthday datetime ,
@phone varchar(50) ,
@lastlogintime datetime,
@status varchar(1) ,
@usercreated varchar(50),
@datecreated datetime ,
@usermodified varchar(50),
@datemodified datetime ,
@islogin varchar(50),
@expiretime datetime ,
@branchid varchar(50),
@deptid varchar(50),
@userlevel int ,
@coreuserid varchar(50),
@corelgnname varchar(255) ,
@productid varchar(50),
@isshow varchar(5) ,
@policyid int ,
@failnumber int

AS
BEGIN

    SET NOCOUNT ON;
    --check duplicate:
    IF EXISTS (SELECT * FROM S_USRAC WHERE usrid=@usrid)
    BEGIN
        DECLARE @ERROR AS VARCHAR(200)
        SELECT @ERROR='USERNAME ' + @usrid + ' Already Existed In Database. Please Input Other Username.'
        RAISERROR (@ERROR ,16,1)
        RETURN;
    END

    SET DATEFORMAT dMy

    INSERT INTO [dbo].[S_USRAC]
    ([username]
    ,[password]
    ,[firstname]
    ,[lastname]
    ,[gender]
    ,[address]
    ,[email]
    ,[birthday]
    ,[phone]
    ,[lastlogintime]
    ,[status]
    ,[usercreated]
    ,[datecreated]
    ,[usermodified]
    ,[datemodified]
    ,[islogin]
    ,[expiretime]
    ,[branchid]
    ,[deptid]
    ,[userlevel]
    ,[coreuserid]
    ,[corelgnname]
    ,[productid]
    ,[isshow]
    ,[policyid]
    ,[failnumber])

    VALUES
    (@username,
    @password,
    @firstname,
    @lastname,
    @gender,
    @address,
    @email,
    @birthday,
    @phone,
    @lastlogintime,
    @status,
    @usercreated,
    @datecreated,
    @usermodified,
    @datemodified,
    @islogin,
    @expiretime,
    @branchid,
    @deptid,
    @userlevel,
    @coreuserid,
    @corelgnname,
    @productid,
```

```
@isshow,  
@policyid,  
@failnumber);
```

END

TRIAL

1.2.4. Procedure: ADM_USER_LOGIN (User Login to System)

Sample field: Status: Active

- This function used to login to System

Input/Output

	Name	Data type	Description
@	username	nvarchar(50)	
@	password	nvarchar(100)	

Uses

Name
ADM_USER_LOGIN (User Login to System)
↳ s_branch (Branch of the bank)
↳ s_department
↳ s_usrac (User Profile)

Script

```
CREATE PROCEDURE [dbo].[ADM_USER_LOGIN]
    @username NVARCHAR(50),
    @password NVARCHAR(100)
AS
BEGIN
    SET NOCOUNT ON;

    SELECT COUNT(a.username)
    FROM s_usrac a LEFT JOIN s_branch b ON b.branchid=a.branchid
    LEFT JOIN s_department c ON c.deptid = a.deptid
    WHERE a.username=@username AND password = @password AND a.status= 'N'
END
```

1.2.5. Procedure: ADM_USER_UPDATE (Update User Profile)

Sample field: Status: Active

- This function used to update one user profile

Input/Output

	Name	Data type	Description
@	username	nvarchar(200)	
@	firstname	nvarchar(200)	
@	lastname	nvarchar(200)	
@	gender	int	
@	address	nvarchar(200)	
@	email	nvarchar(200)	
@	birthday	varchar(10)	
@	phone	nvarchar(20)	
@	status	int	
@	usermodified	varchar(50)	
@	branchid	varchar(20)	
@	userlevel	varchar(10)	
@	policyid	varchar(20)	
@	deptid	varchar(20)	

Uses

Name
ADM_USER_UPDATE (Update User Profile)
S_USRAC (User Profile)

Script

```
CREATE PROCEDURE [dbo].[ADM_USER_UPDATE]
    @username NVARCHAR(200),
    @firstname NVARCHAR(200),
    @lastname NVARCHAR(200),
    @gender INT,
    @address NVARCHAR(200),
    @email NVARCHAR(200),
    @birthday VARCHAR(10),
    @phone NVARCHAR(20),
    @status INT,
    @usermodified VARCHAR(50),
    @branchid VARCHAR(20),
    @userlevel VARCHAR(10),
    @policyid VARCHAR(20),
    @deptid VARCHAR(20)

AS
BEGIN
    SET NOCOUNT ON
    SET DATEFORMAT DMY

    UPDATE [dbo].[S_USRAC] SET
        [firstname] = @firstname,
        [lastname] = @lastname,
        [gender] = @gender,
        [address] = @address,
        [email] = @email,
        [birthday] = cast(@birthday as datetime),
        [phone] = @phone,
        [status] = @status,
        [usermodified] = @usermodified,
        [datemodified] = getdate(),
        [branchid]=@branchid,
        [userlevel]=cast(@userlevel as int),
        policyid=cast(@policyid as int),
        deptid=@deptid
    WHERE
        [username] = @username;
END
```

TRIAL

1.2.6. Procedure: ADM_USERS_SEARCH (Search User Profile)

Sample field: Status: Active

- This function used to Searching one user profile

Input/Output

Name		Data type	Description
@keyword	keyword	nvarchar(50)	
@letter	letter	varchar(5)	

Uses

Name
ADM_USERS_SEARCH (Search User Profile)
S_USRAC (User Profile)

Script

```
CREATE PROCEDURE [dbo].[ADM_USERS_SEARCH]
    @keyword NVARCHAR(50)='',
    @letter VARCHAR(5)
AS
BEGIN
    SET NOCOUNT ON;

    IF (@LETTER=' ')
    BEGIN
        SELECT * FROM S_USRAC WHERE (((USERNAME LIKE '%' +@KEYWORD+'%') OR (@KEYWORD='')) OR ((LASTNAME LIKE
        '%' +@KEYWORD+'%') OR (FIRSTNAME LIKE '%' +@KEYWORD+'%') OR (@KEYWORD=''))) AND USRID NOT IN ('ADMIN','GUEST')
        ORDER BY USERNAME
    END
    ELSE
    BEGIN
        SELECT * FROM S_USRAC
        WHERE ((FIRSTNAME LIKE @LETTER+'%') OR (LASTNAME LIKE @LETTER+'%')) AND USRID NOT IN ('ADMIN','GUEST')
        ORDER BY USERNAME
    END
END
```

2. IPC

2.1. Tables

2.1.1. Table: IPC_SCHEDULEACTION (Schedule Action)

Sample field: Status: Active

- This table stored schedule action call to IPC tran code

Columns

	Name	Data type	Description / Attributes
█	ACTIONID	varchar(50)	
█	ASSEMBLYNAME	varchar(255)	
█	ASSEMBLYTYPE	varchar(255)	
█	METHODNAME	varchar(255)	
█	TPCURL	varchar(255)	Nullable
█	SERVICEID	varchar(50)	
█	WIDGETID	varchar(50)	Nullable
█	DESCRIPTION	nvarchar(MAX)	Nullable
█	HASTABLE	varchar(1)	
█	NUMBERPARA	int	
█	IPCTRANCODE	varchar(50)	Nullable

Unique keys

	Columns	Name / Description
█	ACTIONID	PK_IPC_SCHEDULEACTION

Used By

	Name
█	IPC_SCHEDULEACTION (Schedule Action)
⚙	IPC_SCHEDULELOADACTION

2.1.2. Table: IPC_SCHEDULEDETAIL (Schedule Detail)

Sample field: Status: Active

- This table stored Schedule detail information

Columns

		Name	Data type	Description / Attributes
█	🔑	SCHEDULEID	varchar(50)	
█	🔑	PARANAME	varchar(50)	
█		PARAVALUE	nvarchar(4000)	
█		VALUETYPE	varchar(10)	Nullable Default: 'VALUE'

Unique keys

Columns		Name / Description
█	SCHEDULEID, PARANAME	PK_IPC_SCHEDULEDETAIL

Used By

		Name
█	IPC_SCHEDULEDETAIL (Schedule Detail)	
⚙️	IPC_SCHEDULESGETCURRENT	
⚙️	IPC_SCHEDULESLOGIN	

2.1.3. Table: IPC_SCHEDULEDETAILHIS (Schedule detail history)

Sample field: Status: Active

Columns

	Name	Data type	Description / Attributes
█	SCHEDULEID	varchar(50)	
█	PARANAME	varchar(50)	
█	PARAVALUE	nvarchar(255)	
█	VALUETYPE	varchar(10)	Nullable Default: 'VALUE'

Unique keys

	Columns	Name / Description
█	SCHEDULEID, PARANAME	PK_IPC_SCHEDULEDETAILHIS

Used By

	Name
█	IPC_SCHEDULEDETAILHIS (Schedule detail history)
⚙	IPC_SCHEDULESCURRENT
⚙	IPC_SCHEDULESLOGIN

2.1.4. Table: IPC_SCHEDULEREPEAT (Schedule Repeat)

Sample field: Status: Active

- This table stored schedule repeat.

Columns

	Name	Data type	Description / Attributes
█	SCHEDULEID	nvarchar(50)	
█	DISTANCE	numeric(18, 0)	
█	NRETRY	int	Nullable
█	TRETRY	int	Nullable

Unique keys

	Columns	Name / Description
█	SCHEDULEID	PK_IPC_SCHEDULEREPEAT

Used By

	Name
█	IPC_SCHEDULEREPEAT (Schedule Repeat)
⚙	FAST_SEARCH_SCHEDULE
⚙	FAST_UPDATE_SCHEDULE
⚙	IPC_GETNEXTEXECSCHEDULE
⚙	IPC_SCHEDULESGETCURRENT
⚙	IPC_SCHEDULESLOGIN

2.1.5. Table: IPC_SCHEDULES (Schedule Information)

Sample field: Status: Active

- This table used to stored Schedule information. The Scheduler will process some of task of the system

Columns

	Name	Data type	Description / Attributes
█	SCHEDULEID	varchar(50)	
█	SCHEDULETYPE	varchar(50)	
█	SCHEDULETIME	datetime	
█	SCHEDULENAME	nvarchar(255)	
█	DESCRIPTION	nvarchar(MAX)	Nullable
█	STATUS	varchar(1)	
█	USERCREATE	varchar(50)	
█	USERAPPROVED	varchar(50)	
█	ISAPPROVED	varchar(1)	
█	SERVICEID	varchar(50)	
█	ACTIONTYPE	varchar(50)	
█	ACTIONID	varchar(50)	
█	NEXTEXECUTE	datetime	
█	CREATEDATE	datetime	
█	ENDDATE	datetime	
█	ORD	int	Nullable Default: 0

Unique keys

	Columns	Name / Description
█	SCHEDULEID	PK_IPC_SCHEDULES

Used By

	Name
█	IPC_SCHEDULES (Schedule Information)
⚙	FAST_SEARCH_SCHEDULE
⚙	FAST_UPDATE_SCHEDULE
⚙	IPC_CHECKAPPROVED_SCHEDULE
⚙	IPC_GET_LISTSCHEDULE (Get List Schedule Information)
⚙	IPC_GETNEXTEXECSCHEDULE
⚙	IPC_GETNEXTTIMEEXEC (Get Next Time Execute)
⚙	IPC_GETNEXTTIMEEXEC_bk20171121
⚙	IPC_SCHEDULESGETCURRENT
⚙	IPC_SCHEDULESLOGIN
⚙	IPCSCHEDULE_UPDATENEXTEXEC

2.1.6. Table: IPC_SCHEDULESHIS (Schedule History)

Sample field: Status: Active

Columns

	Name	Data type	Description / Attributes
█	SCHEDULEID	varchar(50)	
█	SCHEDULETYPE	varchar(50)	
█	SCHEDULETIME	datetime	
█	SCHEDULENAME	nvarchar(255)	
█	DESCRIPTION	nvarchar(MAX)	Nullable
█	STATUS	varchar(1)	
█	USERCREATE	varchar(50)	
█	USERAPPROVED	varchar(50)	
█	ISAPPROVED	varchar(1)	
█	SERVICEID	varchar(50)	
█	ACTIONTYPE	varchar(50)	
█	ACTIONID	varchar(50)	
█	NEXTEXECUTE	datetime	
█	CREATEDATE	datetime	
█	ENDDATE	datetime	
█	ORD	int	Nullable
█	LASTEXEC	datetime	

Unique keys

	Columns	Name / Description
█	SCHEDULEID	PK_IPC_SCHEDULESHIS

Used By

	Name
█	IPC_SCHEDULESHIS (Schedule History)
⚙	IPC_CHECKAPPROVED_SCHEDULE
⚙	IPC_SCHEDULESGETCURRENT
⚙	IPC_SCHEDULESLOGIN

2.1.7. Table: IPC_SWIFT_LOGDEFINE (IPC Swift Log Define)

Sample field: Status: Active

This table stored SWIFT define TAG

Columns

	Name	Data type	Description / Attributes
MSGTYPE	varchar(50)		
FIELDSTYLE	nvarchar(50)		
FIELDNAME	nvarchar(1000)		
VALUESTYLE	nvarchar(50)		
VALUENAME	nvarchar(1000)		
FORD	int		
CAPTION	nvarchar(200)		
FORMATTYPE	varchar(10)		
FORMATOBJECT	varchar(1000)		
FORMATFUNCTION	varchar(1000)		
FORMATPARM	varchar(1000)		

Unique keys

	Columns	Name / Description
PK_IPC_SWIFT_LOGDEFINE	FORD	

2.1.8. Table: IPCCOMPONENT (IPC component)

Sample field: Status: Active

- This table stored component process call store proc

Columns

	Name	Data type	Description / Attributes
█	SOURCEID	varchar(20)	
█	DESTID	varchar(20)	
█	IPCTRANCODE	varchar(30)	
█	ASSEMBLYFILE	varchar(50)	
█	ASSEMBLYTITLE	varchar(50)	
█	METHOD	varchar(50)	
█	PARMLIST	varchar(1000)	
█	STATUS	varchar(1)	
█	REVERSAL	bit	
█	ORD	int	
█	SU	int	
█	FA	int	

Unique keys

	Columns	Name / Description
█	SOURCEID, DESTID, IPCTRANCODE, STATUS, REVERSAL, ORD	PK_IPCCOMPONENT

Used By

	Name
█	IPCCOMPONENT (IPC component)
⚙	IPC_GETURL_BYSOURCETRANCODE
⚙	IPCCOMPONENT_SELECT
⚙	IPCCOMPONENTSYN_SELECT

2.1.9. Table: IPCCONNECTIONWS (IPC Connection WebService)

Sample field: Status: Active

- This table stored schedule webservice call to Websevice to process action

Columns

	Name	Data type	Description / Attributes
█	DESTID	varchar(20)	
█	RECEIVERCHANNEL	varchar(20)	
█	IPCTRANCODE	varchar(30)	
█	TARGETNAMESPACE	varchar(255)	
█	HEADERVARIABLENAME	varchar(255)	
█	METHODNAME	varchar(255)	
█	URLWEBSERVICE	varchar(255)	
█	SOAPACTION	varchar(255)	
█	CONTENTTYPE	varchar(255)	Nullable
█	CUSTOMFORMAT	varchar(5)	Nullable Default: 'N'
█	DESCRIPTION	nvarchar(MAX)	
█	WSGRP	varchar(50)	Nullable
█	COREAPI	char(1)	Nullable Default: 'N'

Unique keys

	Columns	Name / Description
█	DESTID, RECEIVERCHANNEL, IPCTRANCODE	PK_IPCCONNECTIONWS

Used By

	Name
█	IPCCONNECTIONWS (IPC Connection WebService)
⚙	IPC_GETURL_BYSOURCETRANCODE

2.1.10. Table: IPCLOGSYNCDEFINE (IPC Log Sync Define)

Sample field: Status: Active

Columns

	Name	Data type	Description / Attributes
PK	IPCTRANCODE	varchar(100)	
PK	DESTID	varchar(30)	
PK	PARMTYPE	varchar(1)	
PK	POS	int	
	FIELDNAME	varchar(50)	
	LOGTYPE	varchar(1)	Nullable
	FORMATTYPE	varchar(10)	Nullable
	FORMATOBJECT	varchar(1000)	Nullable
	FORMATFUNCTION	varchar(1000)	Nullable
	FORMATPARM	varchar(1000)	Nullable

Unique keys

	Columns	Name / Description
PK	IPCTRANCODE, DESTID, PARMTYPE, POS	PK_IPCLOGSYNCDEFINE

2.1.11. Table: IPCOUTPUTDEFINEHTTP (IPC Output Define HTTP)

Sample field: Status: Active

- This table stored detail json tag

Columns

	Name	Data type	Description / Attributes
█	SOURCEID	varchar(20)	
█	DESTID	varchar(20)	
█	IPCTRANCODE	varchar(30)	
█	ONLINE	bit	
█	FIELDNO	int	
█	FIELDDESC	varchar(100)	
█	FIELDSTYLE	varchar(50)	
█	FIELDNAME	nvarchar(1000)	
█	VALUESTYLE	varchar(10)	
█	VALUENAME	nvarchar(1000)	
█	FORMATTYPE	varchar(10)	
█	FORMATOBJECT	varchar(1000)	
█	FORMATFUNCTION	varchar(1000)	
█	FORMATPARM	varchar(1000)	

Unique keys

	Columns	Name / Description
█	SOURCEID, DESTID, IPCTRANCODE, ONLINE, FIELDNO	PK_IPCOUTPUTDEFINEHTTP

2.1.12. Table: IPCSYNCSEQ (IPC SYNC Sequences)

Sample field: Status: Active

- This table stored sync sequence number last transaction

Columns

Name		Data type	Description / Attributes
IPCTRANCODE		varchar(100)	
DESTID		varchar(50)	
SEQNO		numeric(18, 0)	Nullable

Unique keys

Columns		Name / Description
	IPCTRANCODE, DESTID	PK_IPCSYNCSEQ

TRIAL

2.1.13. Table: IPCTRANCODEMAPING_ISO (IPC transacion code mapping ISO)

Sample field: Status: Active

Columns

Name		Data type	Description / Attributes
	IPCTRANCODE	varchar(100)	Nullable
	TRANCODEMAPPING	varchar(100)	Nullable
	TRANTYPE	varchar(10)	Nullable

Used By

Name
IPCTRANCODEMAPING_ISO (IPC transacion code mapping ISO)
FAST_SELECTTRANCODEMAPPINGISO (FAST Get transaction code mapping ISO)

TRIAL

2.1.14. Table: SYSERRORLIST (System Error List)

Sample field: Status: Active

Columns

Name		Data type	Description / Attributes
	IPCTRANCODE	varchar(50)	
	ERRORCODE	varchar(10)	
	ERRORDESC	nvarchar(200)	

Unique keys

Columns		Name / Description
	IPCTRANCODE, ERRORCODE	PK_SYSERRORLIST

TRIAL

2.2. Procedures

2.2.1. Procedure: IPC_GET_LISTSCHEDULE (Get List Schedule Information)

Sample field: Status: Active

- This store procedure used to get list schedule information of system

Uses

Name
 IPC_GET_LISTSCHEDULE (Get List Schedule Information)
 ipc_schedules (Schedule Information)

Script

```
create  procedure [dbo].[IPC_GET_LISTSCHEDULE]
AS
BEGIN
    SET NOCOUNT ON;
    SELECT scheduleid,
           scheduletype,
           schedulename,
           description,
           status,
           usercreate,
           userapproved,
           isapproved,
           serviceid,
           actiontype,
           actionid,
           nextexecute,
           createdate,
           enddate
    FROM ipc_schedules;
END
```

2.2.2. Procedure: IPC_GETNEXTTIMEEXEC (Get Next Time Execute)

Sample field: Status: Active

- This procedure used to get next time execute schedule

Uses

Name
IPC_GETNEXTTIMEEXEC (Get Next Time Execute)
IPC_SCHEDULES (Schedule Information)

Script

```
CREATE PROCEDURE [dbo].[IPC_GETNEXTTIMEEXEC]
AS
BEGIN
    SET NOCOUNT ON;
    UPDATE IPC_SCHEDULES
    SET NEXTEXECUTE=DATEADD(SECOND,3,GETDATE())
    WHERE NEXTEXECUTE <= GETDATE() AND STATUS = 'A'
    DECLARE @MINN DATETIME = (SELECT MIN(NEXTEXECUTE) FROM IPC_SCHEDULES WITH (NOLOCK) WHERE STATUS = 'A')
    SELECT CASE WHEN @MINN > GETDATE() THEN @MINN - GETDATE() ELSE DATEADD(SECOND,3,GETDATE()) - GETDATE() END
NEXTEXECUTE
END
```

TRIAL

2.2.3. Procedure: IPC_UPDATELOGDEFINE_EVENT (IPC Update Log Define Event)

Sample field: Status: Active

Input/Output

	Name	Data type	Description
@	IPCTRANSID	numeric(18, 4)	
@	CHAR01	nvarchar(1000)	
@	CHAR02	nvarchar(1000)	
@	CHAR03	nvarchar(1000)	
@	CHAR04	nvarchar(1000)	
@	CHAR05	nvarchar(1000)	
@	CHAR06	nvarchar(1000)	
@	CHAR07	nvarchar(1000)	
@	CHAR08	nvarchar(1000)	
@	CHAR09	nvarchar(1000)	
@	CHAR10	nvarchar(1000)	
@	CHAR11	nvarchar(1000)	
@	CHAR12	nvarchar(1000)	
@	CHAR13	nvarchar(1000)	
@	CHAR14	nvarchar(1000)	
@	CHAR15	nvarchar(1000)	
@	CHAR16	nvarchar(1000)	
@	CHAR17	nvarchar(1000)	
@	CHAR18	nvarchar(1000)	
@	CHAR19	nvarchar(1000)	
@	CHAR20	nvarchar(1000)	
@	CHAR21	nvarchar(1000)	
@	CHAR22	nvarchar(1000)	
@	CHAR23	nvarchar(1000)	
@	CHAR24	nvarchar(1000)	
@	CHAR25	nvarchar(1000)	
@	CHAR26	nvarchar(1000)	
@	CHAR27	nvarchar(1000)	
@	CHAR28	nvarchar(1000)	
@	CHAR29	nvarchar(1000)	
@	CHAR30	nvarchar(1000)	
@	CHAR31	nvarchar(1000)	
@	CHAR32	nvarchar(1000)	
@	CHAR33	nvarchar(1000)	
@	CHAR34	nvarchar(1000)	

	Name	Data type	Description
@	CHAR35	nvarchar(1000)	
@	CHAR36	nvarchar(1000)	
@	CHAR37	nvarchar(1000)	
@	CHAR38	nvarchar(1000)	
@	CHAR39	nvarchar(1000)	
@	CHAR40	nvarchar(1000)	
@	CHAR41	nvarchar(1000)	
@	CHAR42	nvarchar(1000)	
@	CHAR43	nvarchar(1000)	
@	CHAR44	nvarchar(1000)	
@	CHAR45	nvarchar(1000)	
@	CHAR46	nvarchar(1000)	
@	CHAR47	nvarchar(1000)	
@	CHAR48	nvarchar(1000)	
@	CHAR49	nvarchar(1000)	
@	CHAR50	nvarchar(1000)	
@	CHAR51	nvarchar(1000)	
@	CHAR52	nvarchar(1000)	
@	CHAR53	nvarchar(1000)	
@	CHAR54	nvarchar(1000)	
@	CHAR55	nvarchar(1000)	
@	CHAR56	nvarchar(1000)	
@	CHAR57	nvarchar(1000)	
@	CHAR58	nvarchar(1000)	
@	CHAR59	nvarchar(1000)	
@	CHAR60	nvarchar(1000)	
@	CHAR61	nvarchar(1000)	
@	CHAR62	nvarchar(1000)	
@	CHAR63	nvarchar(1000)	
@	CHAR64	nvarchar(1000)	
@	CHAR65	nvarchar(1000)	
@	CHAR66	nvarchar(1000)	
@	CHAR67	nvarchar(1000)	
@	CHAR68	nvarchar(1000)	
@	CHAR69	nvarchar(1000)	
@	CHAR70	nvarchar(1000)	
@	NUM01	numeric(18, 4)	
@	NUM02	numeric(18, 4)	
@	NUM03	numeric(18, 4)	
@	NUM04	numeric(18, 4)	

	Name	Data type	Description
@	NUM05	numeric(18, 4)	
@	NUM06	numeric(18, 4)	
@	NUM07	numeric(18, 4)	
@	NUM08	numeric(18, 4)	
@	NUM09	numeric(18, 4)	
@	NUM10	numeric(18, 4)	
@	NUM11	numeric(18, 4)	
@	NUM12	numeric(18, 4)	
@	NUM13	numeric(18, 4)	
@	NUM14	numeric(18, 4)	
@	NUM15	numeric(18, 4)	
@	NUM16	numeric(18, 4)	
@	NUM17	numeric(18, 4)	
@	NUM18	numeric(18, 4)	
@	NUM19	numeric(18, 4)	
@	NUM20	numeric(18, 4)	
@	NUM21	numeric(18, 4)	
@	NUM22	numeric(18, 4)	
@	NUM23	numeric(18, 4)	
@	NUM24	numeric(18, 4)	
@	NUM25	numeric(18, 4)	
@	NUM26	numeric(18, 4)	
@	NUM27	numeric(18, 4)	
@	NUM28	numeric(18, 4)	
@	NUM29	numeric(18, 4)	
@	NUM30	numeric(18, 4)	
@	NUM31	numeric(18, 4)	
@	NUM32	numeric(18, 4)	
@	NUM33	numeric(18, 4)	
@	NUM34	numeric(18, 4)	
@	NUM35	numeric(18, 4)	
@	NUM36	numeric(18, 4)	
@	NUM37	numeric(18, 4)	
@	NUM38	numeric(18, 4)	
@	NUM39	numeric(18, 4)	
@	NUM40	numeric(18, 4)	
@	NUM41	numeric(20, 4)	
@	NUM42	numeric(20, 4)	
@	NUM43	numeric(20, 4)	
@	NUM44	numeric(20, 4)	

	Name	Data type	Description
@	NUM45	numeric(20, 4)	
@	NUM46	numeric(20, 4)	
@	NUM47	numeric(20, 4)	
@	NUM48	numeric(20, 4)	
@	NUM49	numeric(20, 4)	
@	NUM50	numeric(20, 4)	
@	NUM51	numeric(20, 4)	
@	NUM52	numeric(20, 4)	
@	NUM53	numeric(20, 4)	
@	NUM54	numeric(20, 4)	
@	NUM55	numeric(20, 4)	
@	NUM56	numeric(20, 4)	
@	NUM57	numeric(20, 4)	
@	NUM58	numeric(20, 4)	
@	NUM59	numeric(20, 4)	
@	NUM60	numeric(20, 4)	
@	NUM61	numeric(20, 4)	
@	NUM62	numeric(20, 4)	
@	NUM63	numeric(20, 4)	
@	NUM64	numeric(20, 4)	
@	NUM65	numeric(20, 4)	
@	NUM66	numeric(20, 4)	
@	NUM67	numeric(20, 4)	
@	NUM68	numeric(20, 4)	
@	NUM69	numeric(20, 4)	
@	NUM70	numeric(20, 4)	

Uses

Name
IPC_UPDATELOGDEFINE_EVENT (IPC Update Log Define Event)
IPCLOGTRANSEVENT

Script

```

CREATE
PROCEDURE [dbo].[IPC_UPDATELOGDEFINE_EVENT]
    @IPCTRANSID NUMERIC(18,4)
    ,@CHAR01 NVARCHAR(1000)
    ,@CHAR02 NVARCHAR(1000)
    ,@CHAR03 NVARCHAR(1000)
    ,@CHAR04 NVARCHAR(1000)
    ,@CHAR05 NVARCHAR(1000)
    ,@CHAR06 NVARCHAR(1000)
    ,@CHAR07 NVARCHAR(1000)
    ,@CHAR08 NVARCHAR(1000)
    ,@CHAR09 NVARCHAR(1000)
    ,@CHAR10 NVARCHAR(1000)
    ,@CHAR11 NVARCHAR(1000)
    ,@CHAR12 NVARCHAR(1000)
    ,@CHAR13 NVARCHAR(1000)
    ,@CHAR14 NVARCHAR(1000)
    ,@CHAR15 NVARCHAR(1000)

```

,@CHAR16 NVARCHAR(1000)
,@CHAR17 NVARCHAR(1000)
,@CHAR18 NVARCHAR(1000)
,@CHAR19 NVARCHAR(1000)
,@CHAR20 NVARCHAR(1000)
,@CHAR21 NVARCHAR(1000)
,@CHAR22 NVARCHAR(1000)
,@CHAR23 NVARCHAR(1000)
,@CHAR24 NVARCHAR(1000)
 ,@CHAR25 NVARCHAR(1000)
 ,@CHAR26 NVARCHAR(1000),
 @CHAR27 NVARCHAR(1000),
 @CHAR28 NVARCHAR(1000),
 @CHAR29 NVARCHAR(1000),
 @CHAR30 NVARCHAR(1000),
 @CHAR31 NVARCHAR(1000),
 @CHAR32 NVARCHAR(1000),
 @CHAR33 NVARCHAR(1000),
 @CHAR34 NVARCHAR(1000),
 @CHAR35 NVARCHAR(1000),
 @CHAR36 NVARCHAR(1000),
 @CHAR37 NVARCHAR(1000),
 @CHAR38 NVARCHAR(1000),
 @CHAR39 NVARCHAR(1000),
 @CHAR40 NVARCHAR(1000),
 @CHAR41 NVARCHAR(1000),
 @CHAR42 NVARCHAR(1000),
 @CHAR43 NVARCHAR(1000),
 @CHAR44 NVARCHAR(1000),
 @CHAR45 NVARCHAR(1000),
 @CHAR46 NVARCHAR(1000),
 @CHAR47 NVARCHAR(1000),
 @CHAR48 NVARCHAR(1000),
 @CHAR49 NVARCHAR(1000),
 @CHAR50 NVARCHAR(1000),
 @CHAR51 NVARCHAR(1000),
 @CHAR52 NVARCHAR(1000),
 @CHAR53 NVARCHAR(1000),
 @CHAR54 NVARCHAR(1000),
 @CHAR55 NVARCHAR(1000),
 @CHAR56 NVARCHAR(1000),
 @CHAR57 NVARCHAR(1000),
 @CHAR58 NVARCHAR(1000),
 @CHAR59 NVARCHAR(1000),
 @CHAR60 NVARCHAR(1000),
 @CHAR61 NVARCHAR(1000),
 @CHAR62 NVARCHAR(1000),
 @CHAR63 NVARCHAR(1000),
 @CHAR64 NVARCHAR(1000),
 @CHAR65 NVARCHAR(1000),
 @CHAR66 NVARCHAR(1000),
 @CHAR67 NVARCHAR(1000),
 @CHAR68 NVARCHAR(1000),
 @CHAR69 NVARCHAR(1000),
 @CHAR70 NVARCHAR(1000),
 ,@NUM01 NUMERIC(18,4)
 ,@NUM02 NUMERIC(18,4)
 ,@NUM03 NUMERIC(18,4)
 ,@NUM04 NUMERIC(18,4)
 ,@NUM05 NUMERIC(18,4)
 ,@NUM06 NUMERIC(18,4)
 ,@NUM07 NUMERIC(18,4)
 ,@NUM08 NUMERIC(18,4)
 ,@NUM09 NUMERIC(18,4)
 ,@NUM10 NUMERIC(18,4)
 ,@NUM11 NUMERIC(18,4)
 ,@NUM12 NUMERIC(18,4)
 ,@NUM13 NUMERIC(18,4)
 ,@NUM14 NUMERIC(18,4)
 ,@NUM15 NUMERIC(18,4)
 ,@NUM16 NUMERIC(18,4)
 ,@NUM17 NUMERIC(18,4)
 ,@NUM18 NUMERIC(18,4)
 ,@NUM19 NUMERIC(18,4)
 ,@NUM20 NUMERIC(18,4)
 ,@NUM21 NUMERIC(18,4)
 ,@NUM22 NUMERIC(18,4)
 ,@NUM23 NUMERIC(18,4)
 ,@NUM24 NUMERIC(18,4)
 ,@NUM25 NUMERIC(18,4),
 @NUM26 NUMERIC(18,4),
 @NUM27 NUMERIC(18,4),
 @NUM28 NUMERIC(18,4),
 @NUM29 NUMERIC(18,4),
 @NUM30 NUMERIC(18,4),
 @NUM31 NUMERIC(18,4),
 @NUM32 NUMERIC(18,4),
 @NUM33 NUMERIC(18,4),
 @NUM34 NUMERIC(18,4),
 @NUM35 NUMERIC(18,4),
 @NUM36 NUMERIC(18,4),
 @NUM37 NUMERIC(18,4),
 @NUM38 NUMERIC(18,4),

```

@NUM39          NUMERIC(18,4),
@NUM40          NUMERIC(18,4),
@NUM41          NUMERIC(20,4),
@NUM42          NUMERIC(20,4),
@NUM43          NUMERIC(20,4),
@NUM44          NUMERIC(20,4),
@NUM45          NUMERIC(20,4),
@NUM46          NUMERIC(20,4),
@NUM47          NUMERIC(20,4),
@NUM48          NUMERIC(20,4),
@NUM49          NUMERIC(20,4),
@NUM50          NUMERIC(20,4),
@NUM51          NUMERIC(20,4),
@NUM52          NUMERIC(20,4),
@NUM53          NUMERIC(20,4),
@NUM54          NUMERIC(20,4),
@NUM55          NUMERIC(20,4),
@NUM56          NUMERIC(20,4),
@NUM57          NUMERIC(20,4),
@NUM58          NUMERIC(20,4),
@NUM59          NUMERIC(20,4),
@NUM60          NUMERIC(20,4),
@NUM61          NUMERIC(20,4),
@NUM62          NUMERIC(20,4),
@NUM63          NUMERIC(20,4),
@NUM64          NUMERIC(20,4),
@NUM65          NUMERIC(20,4),
@NUM66          NUMERIC(20,4),
@NUM67          NUMERIC(20,4),
@NUM68          NUMERIC(20,4),
@NUM69          NUMERIC(20,4),
@NUM70          NUMERIC(20,4)
AS
BEGIN
UPDATE [dbo].[IPCLOGTRANSEVENT]
SET
[CHAR01] = CASE WHEN @CHAR01 = '' THEN CHAR01 ELSE @CHAR01 END
,[CHAR02] = CASE WHEN @CHAR02 = '' THEN CHAR02 ELSE @CHAR02 END
,[CHAR03] = CASE WHEN @CHAR03 = '' THEN CHAR03 ELSE @CHAR03 END
,[CHAR04] = CASE WHEN @CHAR04 = '' THEN CHAR04 ELSE @CHAR04 END
,[CHAR05] = CASE WHEN @CHAR05 = '' THEN CHAR05 ELSE @CHAR05 END
,[CHAR06] = CASE WHEN @CHAR06 = '' THEN CHAR06 ELSE @CHAR06 END
,[CHAR07] = CASE WHEN @CHAR07 = '' THEN CHAR07 ELSE @CHAR07 END
,[CHAR08] = CASE WHEN @CHAR08 = '' THEN CHAR08 ELSE @CHAR08 END
,[CHAR09] = CASE WHEN @CHAR09 = '' THEN CHAR09 ELSE @CHAR09 END
,[CHAR10] = CASE WHEN @CHAR10 = '' THEN CHAR10 ELSE @CHAR10 END
,[CHAR11] = CASE WHEN @CHAR11 = '' THEN CHAR11 ELSE @CHAR11 END
,[CHAR12] = CASE WHEN @CHAR12 = '' THEN CHAR12 ELSE @CHAR12 END
,[CHAR13] = CASE WHEN @CHAR13 = '' THEN CHAR13 ELSE @CHAR13 END
,[CHAR14] = CASE WHEN @CHAR14 = '' THEN CHAR14 ELSE @CHAR14 END
,[CHAR15] = CASE WHEN @CHAR15 = '' THEN CHAR15 ELSE @CHAR15 END
,[CHAR16] = CASE WHEN @CHAR16 = '' THEN CHAR16 ELSE @CHAR16 END
,[CHAR17] = CASE WHEN @CHAR17 = '' THEN CHAR17 ELSE @CHAR17 END
,[CHAR18] = CASE WHEN @CHAR18 = '' THEN CHAR18 ELSE @CHAR18 END
,[CHAR19] = CASE WHEN @CHAR19 = '' THEN CHAR19 ELSE @CHAR19 END
,[CHAR20] = CASE WHEN @CHAR20 = '' THEN CHAR20 ELSE @CHAR20 END
,[CHAR21] = CASE WHEN @CHAR21 = '' THEN CHAR21 ELSE @CHAR21 END
,[CHAR22] = CASE WHEN @CHAR22 = '' THEN CHAR22 ELSE @CHAR22 END
,[CHAR23] = CASE WHEN @CHAR23 = '' THEN CHAR23 ELSE @CHAR23 END
,[CHAR24] = CASE WHEN @CHAR24 = '' THEN CHAR24 ELSE @CHAR24 END
,[CHAR25] = CASE WHEN @CHAR25 = '' THEN CHAR25 ELSE @CHAR25 END
,[CHAR26] = CASE WHEN @CHAR26 = '' THEN CHAR26 ELSE @CHAR26 END
,[CHAR27] = CASE WHEN @CHAR27 = '' THEN CHAR27 ELSE @CHAR27 END
,[CHAR28] = CASE WHEN @CHAR28 = '' THEN CHAR28 ELSE @CHAR28 END
,[CHAR29] = CASE WHEN @CHAR29 = '' THEN CHAR29 ELSE @CHAR29 END
,[CHAR30] = CASE WHEN @CHAR30 = '' THEN CHAR30 ELSE @CHAR30 END
,[CHAR31] = CASE WHEN @CHAR31 = '' THEN CHAR31 ELSE @CHAR31 END
,[CHAR32] = CASE WHEN @CHAR32 = '' THEN CHAR32 ELSE @CHAR32 END
,[CHAR33] = CASE WHEN @CHAR33 = '' THEN CHAR33 ELSE @CHAR33 END
,[CHAR34] = CASE WHEN @CHAR34 = '' THEN CHAR34 ELSE @CHAR34 END
,[CHAR35] = CASE WHEN @CHAR35 = '' THEN CHAR35 ELSE @CHAR35 END
,[CHAR36] = CASE WHEN @CHAR36 = '' THEN CHAR36 ELSE @CHAR36 END
,[CHAR37] = CASE WHEN @CHAR37 = '' THEN CHAR37 ELSE @CHAR37 END
,[CHAR38] = CASE WHEN @CHAR38 = '' THEN CHAR38 ELSE @CHAR38 END
,[CHAR39] = CASE WHEN @CHAR39 = '' THEN CHAR39 ELSE @CHAR39 END
,[CHAR40] = CASE WHEN @CHAR40 = '' THEN CHAR40 ELSE @CHAR40 END
,[CHAR41] = CASE WHEN @CHAR41 = '' THEN CHAR41 ELSE @CHAR41 END
,[CHAR42] = CASE WHEN @CHAR42 = '' THEN CHAR42 ELSE @CHAR42 END
,[CHAR43] = CASE WHEN @CHAR43 = '' THEN CHAR43 ELSE @CHAR43 END
,[CHAR44] = CASE WHEN @CHAR44 = '' THEN CHAR44 ELSE @CHAR44 END
,[CHAR45] = CASE WHEN @CHAR45 = '' THEN CHAR45 ELSE @CHAR45 END
,[CHAR46] = CASE WHEN @CHAR46 = '' THEN CHAR46 ELSE @CHAR46 END
,[CHAR47] = CASE WHEN @CHAR47 = '' THEN CHAR47 ELSE @CHAR47 END
,[CHAR48] = CASE WHEN @CHAR48 = '' THEN CHAR48 ELSE @CHAR48 END
,[CHAR49] = CASE WHEN @CHAR49 = '' THEN CHAR49 ELSE @CHAR49 END
,[CHAR50] = CASE WHEN @CHAR50 = '' THEN CHAR50 ELSE @CHAR50 END
,[CHAR51] = CASE WHEN @CHAR51 = '' THEN CHAR51 ELSE @CHAR51 END
,[CHAR52] = CASE WHEN @CHAR52 = '' THEN CHAR52 ELSE @CHAR52 END
,[CHAR53] = CASE WHEN @CHAR53 = '' THEN CHAR53 ELSE @CHAR53 END
,[CHAR54] = CASE WHEN @CHAR54 = '' THEN CHAR54 ELSE @CHAR54 END
,[CHAR55] = CASE WHEN @CHAR55 = '' THEN CHAR55 ELSE @CHAR55 END
,[CHAR56] = CASE WHEN @CHAR56 = '' THEN CHAR56 ELSE @CHAR56 END

```

```

,[CHAR57] = CASE WHEN @CHAR57 = '' THEN CHAR57 ELSE @CHAR57 END
,[CHAR58] = CASE WHEN @CHAR58 = '' THEN CHAR58 ELSE @CHAR58 END
,[CHAR59] = CASE WHEN @CHAR59 = '' THEN CHAR59 ELSE @CHAR59 END
,[CHAR60] = CASE WHEN @CHAR60 = '' THEN CHAR60 ELSE @CHAR60 END
,[CHAR61] = CASE WHEN @CHAR61 = '' THEN CHAR61 ELSE @CHAR61 END
,[CHAR62] = CASE WHEN @CHAR62 = '' THEN CHAR62 ELSE @CHAR62 END
,[CHAR63] = CASE WHEN @CHAR63 = '' THEN CHAR63 ELSE @CHAR63 END
,[CHAR64] = CASE WHEN @CHAR64 = '' THEN CHAR64 ELSE @CHAR64 END
,[CHAR65] = CASE WHEN @CHAR65 = '' THEN CHAR65 ELSE @CHAR65 END
,[CHAR66] = CASE WHEN @CHAR66 = '' THEN CHAR66 ELSE @CHAR66 END
,[CHAR67] = CASE WHEN @CHAR67 = '' THEN CHAR67 ELSE @CHAR67 END
,[CHAR68] = CASE WHEN @CHAR68 = '' THEN CHAR68 ELSE @CHAR68 END
,[CHAR69] = CASE WHEN @CHAR69 = '' THEN CHAR69 ELSE @CHAR69 END
,[CHAR70] = CASE WHEN @CHAR70 = '' THEN CHAR70 ELSE @CHAR70 END

,[NUM01] = CASE WHEN @NUM01 = 0 THEN NUM01 ELSE @NUM01 END
,[NUM02] = CASE WHEN @NUM02 = 0 THEN NUM02 ELSE @NUM02 END
,[NUM03] = CASE WHEN @NUM03 = 0 THEN NUM03 ELSE @NUM03 END
,[NUM04] = CASE WHEN @NUM04 = 0 THEN NUM04 ELSE @NUM04 END
,[NUM05] = CASE WHEN @NUM05 = 0 THEN NUM05 ELSE @NUM05 END
,[NUM06] = CASE WHEN @NUM06 = 0 THEN NUM06 ELSE @NUM06 END
,[NUM07] = CASE WHEN @NUM07 = 0 THEN NUM07 ELSE @NUM07 END
,[NUM08] = CASE WHEN @NUM08 = 0 THEN NUM08 ELSE @NUM08 END
,[NUM09] = CASE WHEN @NUM09 = 0 THEN NUM09 ELSE @NUM09 END
,[NUM10] = CASE WHEN @NUM10 = 0 THEN NUM10 ELSE @NUM10 END
,[NUM11] = CASE WHEN @NUM11 = 0 THEN NUM11 ELSE @NUM11 END
,[NUM12] = CASE WHEN @NUM12 = 0 THEN NUM12 ELSE @NUM12 END
,[NUM13] = CASE WHEN @NUM13 = 0 THEN NUM13 ELSE @NUM13 END
,[NUM14] = CASE WHEN @NUM14 = 0 THEN NUM14 ELSE @NUM14 END
,[NUM15] = CASE WHEN @NUM15 = 0 THEN NUM15 ELSE @NUM15 END
,[NUM16] = CASE WHEN @NUM16 = 0 THEN NUM16 ELSE @NUM16 END
,[NUM17] = CASE WHEN @NUM17 = 0 THEN NUM17 ELSE @NUM17 END
,[NUM18] = CASE WHEN @NUM18 = 0 THEN NUM18 ELSE @NUM18 END
,[NUM19] = CASE WHEN @NUM19 = 0 THEN NUM19 ELSE @NUM19 END
,[NUM20] = CASE WHEN @NUM20 = 0 THEN NUM20 ELSE @NUM20 END
,[NUM21] = CASE WHEN @NUM21 = 0 THEN NUM21 ELSE @NUM21 END
,[NUM22] = CASE WHEN @NUM22 = 0 THEN NUM22 ELSE @NUM22 END
,[NUM23] = CASE WHEN @NUM23 = 0 THEN NUM23 ELSE @NUM23 END
,[NUM24] = CASE WHEN @NUM24 = 0 THEN NUM24 ELSE @NUM24 END
,[NUM25] = CASE WHEN @NUM25 = 0 THEN NUM25 ELSE @NUM25 END
,[NUM26] = CASE WHEN @NUM26 = 0 THEN NUM26 ELSE @NUM26 END
,[NUM27] = CASE WHEN @NUM27 = 0 THEN NUM27 ELSE @NUM27 END
,[NUM28] = CASE WHEN @NUM28 = 0 THEN NUM28 ELSE @NUM28 END
,[NUM29] = CASE WHEN @NUM29 = 0 THEN NUM29 ELSE @NUM29 END
,[NUM30] = CASE WHEN @NUM30 = 0 THEN NUM30 ELSE @NUM30 END
,[NUM31] = CASE WHEN @NUM31 = 0 THEN NUM31 ELSE @NUM31 END
,[NUM32] = CASE WHEN @NUM32 = 0 THEN NUM32 ELSE @NUM32 END
,[NUM33] = CASE WHEN @NUM33 = 0 THEN NUM33 ELSE @NUM33 END
,[NUM34] = CASE WHEN @NUM34 = 0 THEN NUM34 ELSE @NUM34 END
,[NUM35] = CASE WHEN @NUM35 = 0 THEN NUM35 ELSE @NUM35 END
,[NUM36] = CASE WHEN @NUM36 = 0 THEN NUM36 ELSE @NUM36 END
,[NUM37] = CASE WHEN @NUM37 = 0 THEN NUM37 ELSE @NUM37 END
,[NUM38] = CASE WHEN @NUM38 = 0 THEN NUM38 ELSE @NUM38 END
,[NUM39] = CASE WHEN @NUM39 = 0 THEN NUM39 ELSE @NUM39 END
,[NUM40] = CASE WHEN @NUM40 = 0 THEN NUM40 ELSE @NUM40 END
,[NUM41] = CASE WHEN @NUM41 = 0 THEN NUM41 ELSE @NUM41 END
,[NUM42] = CASE WHEN @NUM42 = 0 THEN NUM42 ELSE @NUM42 END
,[NUM43] = CASE WHEN @NUM43 = 0 THEN NUM43 ELSE @NUM43 END
,[NUM44] = CASE WHEN @NUM44 = 0 THEN NUM44 ELSE @NUM44 END
,[NUM45] = CASE WHEN @NUM45 = 0 THEN NUM45 ELSE @NUM45 END
,[NUM46] = CASE WHEN @NUM46 = 0 THEN NUM46 ELSE @NUM46 END
,[NUM47] = CASE WHEN @NUM47 = 0 THEN NUM47 ELSE @NUM47 END
,[NUM48] = CASE WHEN @NUM48 = 0 THEN NUM48 ELSE @NUM48 END
,[NUM49] = CASE WHEN @NUM49 = 0 THEN NUM49 ELSE @NUM49 END
,[NUM50] = CASE WHEN @NUM50 = 0 THEN NUM50 ELSE @NUM50 END
,[NUM51] = CASE WHEN @NUM51 = 0 THEN NUM51 ELSE @NUM51 END
,[NUM52] = CASE WHEN @NUM52 = 0 THEN NUM52 ELSE @NUM52 END
,[NUM53] = CASE WHEN @NUM53 = 0 THEN NUM53 ELSE @NUM53 END
,[NUM54] = CASE WHEN @NUM54 = 0 THEN NUM54 ELSE @NUM54 END
,[NUM55] = CASE WHEN @NUM55 = 0 THEN NUM55 ELSE @NUM55 END
,[NUM56] = CASE WHEN @NUM56 = 0 THEN NUM56 ELSE @NUM56 END
,[NUM57] = CASE WHEN @NUM57 = 0 THEN NUM57 ELSE @NUM57 END
,[NUM58] = CASE WHEN @NUM58 = 0 THEN NUM58 ELSE @NUM58 END
,[NUM59] = CASE WHEN @NUM59 = 0 THEN NUM59 ELSE @NUM59 END
,[NUM60] = CASE WHEN @NUM60 = 0 THEN NUM60 ELSE @NUM60 END
,[NUM61] = CASE WHEN @NUM61 = 0 THEN NUM61 ELSE @NUM61 END
,[NUM62] = CASE WHEN @NUM62 = 0 THEN NUM62 ELSE @NUM62 END
,[NUM63] = CASE WHEN @NUM63 = 0 THEN NUM63 ELSE @NUM63 END
,[NUM64] = CASE WHEN @NUM64 = 0 THEN NUM64 ELSE @NUM64 END
,[NUM65] = CASE WHEN @NUM65 = 0 THEN NUM65 ELSE @NUM65 END
,[NUM66] = CASE WHEN @NUM66 = 0 THEN NUM66 ELSE @NUM66 END
,[NUM67] = CASE WHEN @NUM67 = 0 THEN NUM67 ELSE @NUM67 END
,[NUM68] = CASE WHEN @NUM68 = 0 THEN NUM68 ELSE @NUM68 END
,[NUM69] = CASE WHEN @NUM69 = 0 THEN NUM69 ELSE @NUM69 END
,[NUM70] = CASE WHEN @NUM70 = 0 THEN NUM70 ELSE @NUM70 END

WHERE IPCTRANSID = @IPCTRANSID
END

```

2.2.4. Procedure: IPCLOGTRANS_UPDATESTATUS (Log transaction Update Status)

Sample field: Status: Active

Input/Output

	Name	Data type	Description
→@	IPCTRANSID	numeric(18, 0)	
→@	STATUS	varchar(10)	

Uses

Name
IPCLOGTRANS_UPDATESTATUS (Log transaction Update Status)
IPCLOGTRANS

Script

```
CREATE PROCEDURE [dbo].[IPCLOGTRANS_UPDATESTATUS]
    @IPCTRANSID
    @STATUS
AS
BEGIN
    SET DATEFORMAT DMY
    UPDATE IPCLOGTRANS
    SET STATUSHIS=STATUS, STATUS = @STATUS
    WHERE IPCTRANSID = @IPCTRANSID
END
```

2.2.5. Procedure: IPCLOGTRANS_UPDATERSTATUS_SYNCTOCORE (Log transaction update status sync to core)

Sample field: Status: Active

Input/Output

	Name	Data type	Description
@IPCTRANSID	IPCTRANSID	numeric(18, 0)	
@NUM70	NUM70	numeric(18, 0)	
@CHAR48	CHAR48	varchar(200)	

Uses

Name
IPCLOGTRANS_UPDATERSTATUS_SYNCTOCORE (Log transaction update status sync to core)
IPCLOGTRANS

Script

```

CREATE PROCEDURE [dbo].[IPCLOGTRANS_UPDATERSTATUS_SYNCTOCORE]
    @IPCTRANSID           NUMERIC(18, 0),
    @NUM70                 --- NUM70=1 -> SYNC TO CORE, 2 NOT SYNC TO CORE
    @CHAR48                VARCHAR(200)
AS
BEGIN
    SET DATEFORMAT DMY
    DECLARE @TOPLINE AS VARCHAR(200)
    SELECT @TOPLINE=SUBSTRING(CHAR49,1,CASE WHEN CHARINDEX(CHAR(13),CHAR49,1)=0 THEN LEN(CHAR49) ELSE
CHARINDEX(CHAR(13),CHAR49,1)-1 END ) FROM IPCLOGTRANS WHERE IPCTRANSID=@IPCTRANSID
    DECLARE @LENCHAR49 AS INT
    SELECT @LENCHAR49=LEN(CHAR49)   FROM IPCLOGTRANS WHERE IPCTRANSID=@IPCTRANSID
    IF (@LENCHAR49>500) UPDATE IPCLOGTRANS SET CHAR49=SUBSTRING(CHAR49,1,500) WHERE IPCTRANSID=@IPCTRANSID
    IF EXISTS (SELECT * FROM IPCLOGTRANS WHERE IPCTRANSID=@IPCTRANSID AND CHAR57='GDDS' AND CHARINDEX('AAA',CHAR64,1)>0 AND
IPCTRANCODE='FAST_OUTGOING')
        UPDATE IPCLOGTRANS
        SET NUM70=2,NUM69=0,CHAR49=CASE WHEN 'FAST-' +@CHAR48=@TOPLINE THEN CHAR49
                                         ELSE CASE WHEN @TOPLINE='' THEN 'FAST-' +@CHAR48 +
                                             ELSE 'FAST-' +@CHAR48 +
                                             END,NUM67=0,NUM68=2
        +@CHAR48
        CHAR(13) + CHAR49 END
        WHERE IPCTRANSID = @IPCTRANSID
        ELSE
            UPDATE IPCLOGTRANS
            SET NUM70=@NUM70,CHAR49=CASE WHEN 'FAST-' +@CHAR48=@TOPLINE THEN CHAR49
                                         ELSE CASE WHEN @TOPLINE='' THEN 'FAST-' +@CHAR48 +
                                             ELSE 'FAST-' +@CHAR48 +
                                             END,NUM67=0,NUM68=2
            +@CHAR48
            CHAR(13) + CHAR49 END
            WHERE IPCTRANSID = @IPCTRANSID
END

```

3. SYSTEM

3.1. Tables

3.1.1. Table: S_BANK (Bank Information)

Sample field: Status: Active

- This table stored bank information

Columns

	Name	Data type	Description / Attributes
█	bankid	int	Identity / Auto increment
█	bankname	varchar(255)	
█	bankcode	varchar(50)	
█	serviceid	varchar(10)	Nullable Default: 'BLS'
█	uri	varchar(255)	
█	mt103acceptpath	varchar(255)	
█	mt103rejectpath	varchar(255)	
█	status	varchar(10)	
█	usercreated	varchar(50)	Nullable
█	datecreated	datetime	Nullable
█	usermodified	varchar(50)	Nullable
█	datemodified	datetime	Nullable
█	issender	varchar(3)	Nullable Default: 'N'
█	countrycode	varchar(3)	Nullable
█	autogetquote	varchar(5)	Nullable Default: 'N'
█	autoacceptquote	varchar(5)	Nullable Default: 'N'

Unique keys

	Columns	Name / Description
█	bankid	PK_S_BANK

Used By

	Name
█	S_BANK (Bank Information)
⚙	SYS_GET_BANK (Get Bank List)

3.1.2. Table: S_BRANCH (Branch of the bank)

Sample field: Status: Active

- This table is show branch of the bank

Columns

	Name	Data type	Description / Attributes
█	BRANCHID	int	
█	BRANCHCD	nvarchar(5)	
█	REFID	nvarchar(15)	Nullable
█	BRNAME	nvarchar(500)	Nullable
█	ADDRESS	nvarchar(500)	Nullable
█	PHONE	nvarchar(255)	Nullable
█	MPHONE	nvarchar(255)	Nullable
█	TAXCODE	nvarchar(50)	Nullable
█	BCYCID	nchar(3)	Nullable
█	BCYNM	nchar(3)	Nullable
█	LCYCID	nchar(3)	Nullable
█	LCYNM	nchar(3)	Nullable
█	REFCODE	nvarchar(250)	Nullable
█	COUNTRY	nchar(2)	Nullable
█	LANG	nchar(2)	Nullable
█	TIMEZN	int	Nullable
█	NUMFMTT	nchar(1)	Nullable
█	NUMFMTD	nchar(1)	Nullable
█	DATEFMT	nvarchar(10)	Nullable
█	LDATEFMT	nvarchar(10)	
█	TIMEFMT	nvarchar(10)	Nullable
█	ISONLINE	nchar(1)	Nullable
█	UDFIELD1	nvarchar(MAX)	Nullable

Linked from

Table	Join	Title / Name / Description
S_USRAC (User Profile)	S_BRANCH BRANCHID = S_USRACbranchid	fk_S_BRANCH_S_USRAC

Unique keys

Columns	Name / Description
█ BRANCHID	PK_S_BRANCH

Used By

Name
█ S_BRANCH (Branch of the bank)
█ ADM_USER_LOGIN (User Login to System)

Name

 S_USRAC (User Profile)

TRIAL

3.1.3. Table: S_PARAM (System parameter)

Sample field: Status: Active

- This table store some parameter of system

Columns

	Name	Data type	Description / Attributes
█	parcode	nchar(10)	
█	pargrp	nvarchar(50)	
█	🔑 parname	nvarchar(50)	
█	parval	nvarchar(MAX)	Nullable
█	mval	nvarchar(MAX)	Nullable
█	descr	nvarchar(250)	Nullable

Unique keys

	Columns	Name / Description
🔑	parname	PK_S_PARAM

Used By

	Name
█	S_PARAM (System parameter)
⚙️	IPCLOGTRANS_GETTRANSACTION_TOFAST (Get transaction send to FAST)

3.1.4. Table: SYSERRORLISTSOURCE (System error list source)

Sample field: Status: Active

Columns

	Name	Data type	Description / Attributes
	SOURCEID	varchar(20)	
	IPCTRANCODE	varchar(20)	
	SYSERRORCODE	varchar(10)	
	SOURCEERRORCODE	varchar(MAX)	
	SOURCEERRORDESC	varchar(MAX)	

TRIAL

3.2. Procedures

3.2.1. Procedure: SYS_GET_BANK (Get Bank List)

Sample field: Status: Active

- This stored get bank list

Uses

Name
SYS_GET_BANK (Get Bank List)
S_BANK (Bank Information)

Script

```
CREATE PROC [dbo].[SYS_GET_BANK]
AS
BEGIN
    SELECT bankid
    ,bankname
    ,bankcode
    ,serviceid
    ,uri
    ,mt103acceptpath
    ,mt103rejectpath
    ,status
    ,usercreated
    ,(CONVERT(VARCHAR(10), datecreated, 103) + ' ' + CONVERT(VARCHAR(8), datecreated, 108)) AS datecreated
    ,usermodified
    ,(CONVERT(VARCHAR(10), datemodified, 103) + ' ' + CONVERT(VARCHAR(8), datemodified, 108)) AS datemodified
    ,issender
    ,countrycode
    ,autogetquote
    ,autoacceptquote
    FROM S_BANK
END
```

4. FAST

4.1. Tables

4.1.1. Table: FAST_BANK (FAST Bank)

Sample field: Status: Active

- This table stored Bank join FAST

Columns

		Name	Data type	Description / Attributes
█	🔑	bankid	varchar(100)	
█		bankname	varchar(500)	
█		bankstreet	varchar(500)	
█		bankbuilding	varchar(200)	
█		bankpostalcode	varchar(200)	
█		banktown	varchar(200)	
█		bankcountry	varchar(200)	
█		username	varchar(200)	
█		password	varchar(200)	
█		participatecode	varchar(200)	

Unique keys

	Columns	Name / Description
🔑	bankid	PK_FAST_BANK

Used By

	Name
█	FAST_BANK (FAST Bank)
⚙️	FAST_BANKLIST (FAST Bank List)

4.2. Procedures

4.2.1. Procedure: FAST_BANKLIST (FAST Bank List)

Sample field: Status: Active

- This stored get bank list join to FAST

Uses

Name
FAST_BANKLIST (FAST Bank List)
FAST_BANK (FAST Bank)

Script

```
CREATE
PROC dbo.FAST_BANKLIST
AS
SELECT bankid
,bankname
,bankstreet
,bankbuilding
,bankpostalcode
,banktown
,bankcountry
,username
,password
,participatecode
FROM dbo.FAST_BANK
```

4.2.2. Procedure: FAST_GETOUTGOING (FAST GET Outgoing Transaction)

Sample field: Status: Active

- This stored get outgoing transaction information

Uses

Name
FAST_GETOUTGOING (FAST GET Outgoing Transaction)
ipclogtrans
S_SYSTEMCODE

Script

```
create PROCEDURE [dbo].[FAST_GETOUTGOING]
AS
BEGIN
    SELECT      A.IPCTRANSID,
                A.TRANREF,
                A.IPCTRANSDATE,
                A.CHAR01   SACCNO,
                A.CHAR02   SDNAME,
                A.CHAR03   SDCCOUNTRY,
                A.CHAR04   SDTOWN,
                A.CHAR05   SDPCODE,
                A.CHAR06   SDSTREET,
                A.CHAR07   SDBDMB,
                A.CHAR08   SBANK,
                A.CHAR09   SBNAME,
                A.CHAR10   SBCCOUNTRY,
                A.CHAR11   SBTOWN,
                A.CHAR12   SBPCODE,
                A.CHAR13   SBSTREET,
                A.CHAR14   SBDLMB,
                A.CHAR15   SCCR,
                A.CHAR16   RACCNO,
                A.CHAR17   RCNAME,
                A.CHAR18   RCCCOUNTRY,
                A.CHAR19   RCTOWN,
                A.CHAR20   RCSTREET,
                A.CHAR21   RCBDMB,
                A.CHAR22   RBANK,
                A.CHAR23   RBNAME,
                A.CHAR24   RBTOWN,
                A.CHAR25   RCCR,
                A.CHAR26   PMTMTD,
                A.CHAR27   PMTINFID,
                A.CHAR28   TXREFID,
                A.CHAR29   FCNTXREFID,
                A.CHAR30   FCNSTS,
                A.CHAR31   FCNSTSUD,
                A.CHAR32   INSTRID,
                A.CHAR33   CORESTS,
                A.CHAR34   CORESTSUD,
                A.CHAR35   CHRGBR,
                A.CHAR36   BACTHBOOKG,
                A.CHAR37   PURPOSE,
                A.CHAR38   STRINFOR,
                A.CHAR39   CREATEDT,
                A.CHAR40   REFNB,
                A.CHAR41   REFCD,
                A.CHAR42   REQUESTDT,
                A.CHAR43   ENDTOENDID,
                A.CHAR44   MSGID,
                A.STATUS + ' - ' + (SELECT cdname FROM S_SYSTEMCODE WHERE cdname = 'FASTSTS' AND cdid= A.STATUS) AS NEWSTATUS
    FROM dbo.ipclogtrans A
END
```

4.2.3. Procedure: FAST_GETTRANBYFCNTXREFID (Get Transaction By FCN)

Sample field: Status: Active

Input/Output

	Name	Data type	Description
→@	FCNTXREFID	varchar(100)	
→@	TRANCODE	varchar(100)	

Uses

Name
⚙ FAST_GETTRANBYFCNTXREFID (Get Transaction By FCN)
IPCLOGTRANS

Script

```
CREATE PROCEDURE [dbo].[FAST_GETTRANBYFCNTXREFID]
@FCNTXREFID VARCHAR(100),
@TRANCODE VARCHAR(100)
AS
BEGIN
    SELECT * FROM dbo.IPCLOGTRANS (NOLOCK) WHERE IPCTRANCODE = @TRANCODE AND CHAR49=@FCNTXREFID
END
```

4.2.4. Procedure: FAST_GETTRANBYPMTINFID (Get Transaction Payment ID)

Sample field: Status: Active

- This stored get transaction payment ID

Input/Output

Name	Data type	Description
@FCNTXREFID	varchar(100)	
@TRANCODE	varchar(100)	

Uses

Name
FAST_GETTRANBYPMTINFID (Get Transaction Payment ID)
IPCLOGTRANS

Script

```
CREATE PROCEDURE [dbo].[FAST_GETTRANBYPMTINFID]
@FCNTXREFID VARCHAR(100),
@TRANCODE VARCHAR(100)
AS
BEGIN
    SELECT * FROM dbo.IPCLOGTRANS (NOLOCK) WHERE IPCTRANCODE = @TRANCODE AND CHAR47=@FCNTXREFID
END
```

4.2.5. Procedure: FAST_GETTRANSDetail (FAST Get transaction detail)

Sample field: Status: Active

- This stored process get transaction detail by TRANSID

Input/Output

	Name	Data type	Description
@	TRANID	varchar(100)	

Uses

	Name
⚙ FAST_GETTRANSDetail (FAST Get transaction detail)	
↳ ipclogtrans	
↳ S_SYSTEMCODE	

Script

```

CREATE
PROCEDURE [dbo].[FAST_GETTRANSDetail]
    @TRANID VARCHAR(100)
AS
BEGIN
    SELECT      A.IPCTRANSID,
                A.TRANREF,
                A.IPCTRANSDATE,
                A.CHAR01    SACCNO,
                A.CHAR02    SDNAME,
                A.CHAR03    SDCCOUNTRY,
                A.CHAR04    SDTOWN,
                A.CHAR05    SDPCODE,
                A.CHAR06    SDSTREET,
                A.CHAR07    SDBDMB,
                A.CHAR08    SBANK,
                A.CHAR09    SBNAME,
                A.CHAR10    SBCOUNTRY,
                A.CHAR11    SBTOWN,
                A.CHAR12    SEPCODE,
                A.CHAR13    SBSTREET,
                A.CHAR14    SBDMB,
                A.CHAR15    SCCR,
                A.CHAR16    RACCCNO,
                A.CHAR17    RCNAME,
                A.CHAR18    RCCCOUNTRY,
                A.CHAR19    RCTOWN,
                A.CHAR20    RCSTREET,
                A.CHAR21    RCBDMB,
                A.CHAR22    RBANK,
                A.CHAR23    RBNAME,
                A.CHAR24    RBTOWN,
                A.CHAR25    RCCR,
                A.CHAR26    PMTMFD,
                A.CHAR27    PMTINFID,
                A.CHAR28    TXREFID,
                A.CHAR29    FCNTXREFID,
                A.CHAR30    FCNSTS,
                A.CHAR31    FCNSTSUD,
                A.CHAR32    INSTRID,
                A.CHAR33    CORESTS,
                A.CHAR34    CORESTSUD,
                A.CHAR35    CHRGBR,
                A.CHAR36    BATHBOOKG,
                A.CHAR37    PURPOSE,
                A.CHAR38    STRINFOR,
                A.CHAR39    CREATEDT,
                A.CHAR40    REFNB,
                A.CHAR41    REFCD,
                A.CHAR42    REQUESTDT,
                A.CHAR43    ENDTOENDID,
                A.CHAR44    MSGID,
                A.STATUS + ' - ' + (SELECT cdname FROM S_SYSTEMCODE WHERE cdname ='FASTSTS' AND cdid= A.STATUS) AS NEWSTATUS
            FROM dbo.ipclogtrans A WHERE ipctransid = @tranid
END

```

4.2.6. Procedure: FAST_SELECTTRANBYPAYMENTID (FAST Select transaction payment ID)

Sample field: Status: Active

Input/Output

	Name	Data type	Description
→@	MSGID	varchar(255)	
→@	PMTINFID	varchar(255)	
→@	TRANTYPE	varchar(100)	

Uses

Name
⚙️ FAST_SELECTTRANBYPAYMENTID (FAST Select transaction payment ID)
IPCLOGTRANS

Script

```
CREATE PROC [dbo].[FAST_SELECTTRANBYPAYMENTID]
    @MSGID VARCHAR(255),
    @PMTINFID VARCHAR(255),
    @TRANTYPE VARCHAR(100)
AS
BEGIN

    IF (@TRANTYPE='I')
    BEGIN
        SELECT TOP 1 * FROM DBO.IPCLOGTRANS WHERE CHAR67=@PMTINFID AND CHAR54='Y' AND IPCTRANCODE='FAST_INCOMING'
    END
    ELSE
    BEGIN
        SELECT TOP 1 * FROM DBO.IPCLOGTRANS WHERE CHAR67=@PMTINFID AND IPCTRANCODE='FAST_OUTGOING'
    END
END
```

4.2.7. Procedure: FAST_SELECTTRANCODEMAPPINGISO (FAST Get transaction code mapping ISO)

Sample field: Status: Active

- This store procedure process get transaction code mapping ISO

Input/Output

	Name	Data type	Description
→@	TRANCODE	varchar(100)	
→@	TRANTYPE	varchar(10)	

Uses

	Name
⚙️	FAST_SELECTTRANCODEMAPPINGISO (FAST Get transaction code mapping ISO)
💻	IPCTRANCODEMAPING_ISO (IPC transacion code mapping ISO)

Script

```
CREATE PROCEDURE [dbo].[FAST_SELECTTRANCODEMAPPINGISO]
    @TRANCODE VARCHAR(100),
    @TRANTYPE VARCHAR(10)
AS
BEGIN
    SELECT * FROM IPCTRANCODEMAPING_ISO WHERE IPCTRANCODE=@TRANCODE AND TRANTYPE=@TRANTYPE
END
```

4.2.8. Procedure: IPCLOGTRANS_GETTRANSACTION_TOFAST (Get transaction send to FAST)

Sample field: Status: Active

- This store procedure used to execute get transaction send to FAST

Input/Output

	Name	Data type	Description
→@	STATUS	varchar(20)	
→@	IPCTRANCODE	varchar(200)	

Uses

Name
IPCLOGTRANS_GETTRANSACTION_TOFAST (Get transaction send to FAST)
IPCLOGTRANS
S_PARAM (System parameter)
Split

Script

```
CREATE PROCEDURE [dbo].[IPCLOGTRANS_GETTRANSACTION_TOFAST]
    @STATUS VARCHAR(20),
    @IPCTRANCODE VARCHAR(200)

    /*
    TYPE VALUE:
    0: MAKE FULL FUN
    2: MAKE ACK
    3: MAKE REFUN
    4: CHUA DUNG
    */
AS
BEGIN
    -- SET NOCOUNT ON ADDED TO PREVENT EXTRA RESULT SETS FROM
    -- INTERFERING WITH SELECT STATEMENTS.
    SET NOCOUNT ON;
    DECLARE @TRETRYFAST AS INT
    SET @TRETRYFAST=(SELECT parval FROM S_PARAM WHERE parname = 'TRETRYFAST' AND parcode = '09' AND pargrp = 'FAST')

    SELECT * FROM IPCLOGTRANS WHERE NUM68=1 AND IPCTRANCODE=@IPCTRANCODE AND STATUS IN (SELECT ITEM FROM
    SPLIT(@STATUS,'/')) AND NUM67<@TRETRYFAST
    --UPDATE TOTAL RETRY
    IF EXISTS( SELECT * FROM IPCLOGTRANS WHERE NUM68=1 AND IPCTRANCODE=@IPCTRANCODE AND STATUS IN (SELECT ITEM FROM
    SPLIT(@STATUS,'/')) AND NUM67<@TRETRYFAST)
    BEGIN
        UPDATE IPCLOGTRANS SET NUM67=NUM67+1 WHERE NUM68=1 AND IPCTRANCODE=@IPCTRANCODE AND STATUS IN (SELECT ITEM FROM
        SPLIT(@STATUS,'/')) AND NUM67<@TRETRYFAST
    END
END
```

5. Other

5.1. Tables

5.1.1. Table: EBA_ROLES

Columns

	Name	Data type	Description / Attributes
█	ROLEID	int	Identity / Auto increment
█	ROLENAMES	nvarchar(1000)	
█	ROLEDESCRIPTION	nvarchar(4000)	Nullable
█	USERTYPE	varchar(50)	Nullable
█	CONTRACTNO	varchar(50)	Nullable
█	USERCREATED	varchar(50)	Nullable
█	DATECREATED	datetime	Nullable Default: getdate()
█	USERMODIFIED	varchar(50)	Nullable
█	DATEMODIFIED	datetime	Nullable
█	SERVICEID	varchar(50)	Nullable
█	STATUS	varchar(50)	Nullable
█	ISSHOW	varchar(5)	Nullable Default: 'Y'

Unique keys

	Columns	Name / Description
█	ROLEID	PK_SmartPortal_Roles

Used By

	Name
█	EBA_ROLES
█	USERS_GETROLE

5.1.2. Table: EBA_TRANRIGHT

Columns

Name		Data type	Description / Attributes
	ROLEID	int	
	PAGEID	int	
	TRANCODE	varchar(50)	Nullable

Unique keys

Columns		Name / Description
	ROLEID, PAGEID	PK_SMP_PageRight

Used By

Name	
	EBA_TRANRIGHT
	PAGES_CHECKPERMIT

TRIAL

5.1.3. Table: IPC_MAPPING_SWIFT

Columns

	Name	Data type	Description / Attributes
	MSGTYPE	varchar(10)	
	BLOCK	varchar(2)	
	FIELDNAME	varchar(50)	
	FIELDSTYLE	varchar(100)	
	TAGNAME	varchar(50)	
	TAGFORMAT	varchar(50)	
	SEQINLINE	int	
	SEQLINE	int	
	FLDCAPTION	nvarchar(200)	
	TAGORD	int	
	NEWLINE	varchar(1)	
	MAXLENGTH	int	
	PREFIX	nvarchar(50)	
	FORMATTYPE	varchar(20)	
	FORMATOBJECT	varchar(1000)	
	FORMATFUNCTION	varchar(1000)	
	FORMATPARM	varchar(1000)	

Unique keys

	Columns	Name / Description
	SEQINLINE, SEQLINE, TAGORD	PK_KRC_MAPPING_SWIFT

5.1.4. Table: IPC_SCHEDULEACTIONPARA

Columns

	Name	Data type	Description / Attributes
█	ACTIONID	varchar(50)	
█	PARANAME	varchar(50)	
█	PARATYPE	varchar(50)	
█	ISREQUIRE	varchar(1)	
█	DESCRIPTION	nvarchar(MAX)	Nullable

Unique keys

	Columns	Name / Description
█	ACTIONID, PARANAME	PK_IPC_SCHEDULEACTIONPARA

TRIAL

5.1.5. Table: IPC_SCHEDULEDAY

Columns

Name		Data type	Description / Attributes
	SCHEDULEID	varchar(50)	
	DAYNO	int	

Used By

Name
IPC_SCHEDULEDAY
IPC_GETNEXTEXECSCHEDULE
IPC_SCHEDULESGETCURRENT
IPC_SCHEDULESLOGIN

TRIAL

5.1.6. Table: IPC_SCHEDULEDAYHIS

Columns

Name		Data type	Description / Attributes
	SCHEDULEID	varchar(50)	
	DAYNO	int	

Used By

Name
IPC_SCHEDULEDAYHIS
IPC_SCHEDULESGETCURRENT
IPC_SCHEDULESLOGIN

TRIAL

5.1.7. Table: IPC_SCHEDULEREPEATHIS

Columns

Name		Data type	Description / Attributes
	SCHEDULEID	nvarchar(50)	
	DISTANCE	numeric(18, 0)	
	NRETRY	int	Nullable
	TRETRY	int	Nullable

Unique keys

Columns	Name / Description
SCHEDULEID	PK_IPC_SCHEDULEREPEATHIS

Used By

Name	
	IPC_SCHEDULEREPEATHIS
	IPC_SCHEDULESGETCURRENT
	IPC_SCHEDULESLOGIN

5.1.8. Table: IPC_TIMEOUTTRANS

Columns

Name		Data type	Description / Attributes
IPCTRANCODE		varchar(50)	
TRANSDESC		varchar(200)	Nullable
SERVICEID		varchar(50)	
TIMEOUT		int	Nullable

Unique keys

Columns	Name / Description
IPCTRANCODE, SERVICEID	PK_IPC_TIMEOUTTRANS

Used By

Name
IPC_TIMEOUTTRANS
IPC_TIMEOUTTRANS_SELECT
IPCLOGTRANS_TIMEOUT

5.1.9. Table: IPCBLSCHECKREVERSAL

Columns

	Name	Data type	Description / Attributes
█	IPCTRANCODE	varchar(50)	
█	IPCTRANCODEDEST	varchar(50)	
█	DESTID	varchar(50)	
█	CORPID	varchar(50)	
█	SERID	varchar(50)	
█	ERRORCODE	varchar(50)	
█	TYPE	varchar(10)	
█	DESCRIPTION	nvarchar(200)	Nullable

Unique keys

	Columns	Name / Description
█	IPCTRANCODE, IPCTRANCODEDEST, DESTID, CORPID, SERID	PK_IPCCHECKREVERSAL

Used By

	Name
█	IPCBLSCHECKREVERSAL
⚙	IPC_CHECKDESTTRANS

5.1.10. Table: IPCBLSREVERSALFLOW

Columns

	Name	Data type	Description / Attributes
█	IPCTRANCODE	varchar(50)	
█	IPCTRANCODEDEST	varchar(50)	
█	DESTID	varchar(50)	
█	CORPID	varchar(50)	
█	SERID	varchar(50)	
█	ORD	int	
█	RVSTRANCODE	varchar(50)	
█	RVSDESTTRANCODE	varchar(50)	

Unique keys

	Columns	Name / Description
█	IPCTRANCODE, IPCTRANCODEDEST, DESTID, CORPID, SERID, ORD	PK_IPCREVERSALFLOW

Used By

	Name
█	IPCBLSREVERSALFLOW
⚙	IPC_CHECKDESTTRANS

5.1.11. Table: IPCCCONNECTIONDB

Columns

Name		Data type	Description / Attributes
DESTID	varchar(20)		
CONNNAME	varchar(50)	Nullable	
CONNSTRING	varchar(500)	Nullable	
TYPE	varchar(100)	Nullable	

Unique keys

Columns		Name / Description
DESTID		PK_IPCCONNECTIONDB

TRIAL

5.1.12. Table: IPCDATADEFINE

Columns

	Name	Data type	Description / Attributes
█	SOURCEID	varchar(20)	
█	IPCTRANCODE	varchar(20)	
█	FIELDNAME	nvarchar(200)	
█	VALUESTYLE	varchar(10)	
█	VALUENAME	nvarchar(1000)	
█	FORMATTYPE	varchar(10)	
█	FORMATOBJECT	varchar(1000)	
█	FORMATFUNCTION	varchar(1000)	
█	FORMATPARM	varchar(1000)	

Unique keys

	Columns	Name / Description
█	SOURCEID, IPCTRANCODE, FIELDNAME	PK_IPCDATADEFINE

5.1.13. Table: IPCINPUTDEFINEHTTP

Columns

	Name	Data type	Description / Attributes
█	IPCTRANCODE	varchar(50)	
█	IPCDESTTRANCODE	varchar(50)	
█	TAGNAME	varchar(100)	
█	KEYNAME	varchar(100)	
█	POS	int	Nullable
█	FORMAT	varchar(50)	
█	DESCRIPTION	nvarchar(MAX)	Nullable
█	DEFAULTVALUE	nvarchar(50)	Nullable
█	FORMATTYPE	varchar(20)	Nullable
█	FORMATOBJECT	varchar(1000)	Nullable
█	FORMATFUNCTION	varchar(1000)	Nullable
█	FORMATPARM	varchar(1000)	Nullable

Unique keys

	Columns	Name / Description
█	IPCTRANCODE, IPCDESTTRANCODE, KEYNAME	PK_IPCINPUTDEFINEHTTP

5.1.14. Table: IPCINPUTDEFINEXML

Columns

	Name	Data type	Description / Attributes
1	IPCTRANCODE	varchar(50)	
2	IPCDESTTRANCODE	varchar(50)	
3	TAGNAME	varchar(50)	
4	KEYNAME	varchar(50)	
5	POS	int	Nullable
6	FORMAT	varchar(50)	
7	DESCRIPTION	nvarchar(MAX)	Nullable
8	TAGATTRIBUTES	varchar(200)	Nullable
9	TAGGOTATTRIBUTES	bit	Default: 0
10	ARRAYLEVEL	int	Default: 0
11	SUBLEVEL	int	Default: 0
12	ARRAYNODE	varchar(100)	Nullable

5.1.15. Table: IPCINPUTDEFINEXML_ISO_MAPTRANCODE

Columns

	Name	Data type	Description / Attributes
1	IPCTRANCODE	varchar(100)	
2	XMLNS	varchar(500)	
3	REFTRANCODE	varchar(100)	
4	DESCRIPTION	varchar(200)	Nullable

TRIAL

5.1.16. Table: IPCINVENTORY

Columns

Name		Data type	Description / Attributes
	SEQNAME	varchar(50)	
	SEQVALUE	numeric(18, 0)	
	SEQMAX	numeric(18, 0)	

Unique keys

Columns		Name / Description
	SEQNAME	PK_IPCINVENTORY

Used By

Name	
	IPCINVENTORY
	IPC_RESETID
	IPCGETSEQVALUE
	IPCGETSEQVALUE_SCALAR

5.1.17. Table: IPCLOGDEFINE

Columns

Name		Data type	Description / Attributes
IPCTRANCODE	PK	varchar(20)	
PARMTYPE	PK	varchar(1)	
POS	PK	int	
FIELDNAME		varchar(50)	
LOGTYPE	PK	varchar(1)	

Unique keys

Columns	Name / Description
IPCTRANCODE, PARMTYPE, POS, LOGTYPE	PK_IPCLOGDEFINE_1

TRIAL

5.1.18. Table: IPCLOGMESSAGE

Columns

	Name	Data type	Description / Attributes
☰	ID	numeric(18, 0)	Identity / Auto increment
☰	IPCTRANSDATE	datetime	Default: getdate()
☰	IPCWORKDATE	datetime	
☰	IPCTRANSID	numeric(18, 0)	
☰	IPCLOGTYPE	char(2)	
☰	MESSAGETYPE	varchar(10)	
☰	MESSAGE	nvarchar(MAX)	

Unique keys

	Columns	Name / Description
🔑	ID	PK_IPCLOGMESSAGE_1

Used By

	Name
☰	IPCLOGMESSAGE
⚙️	IPCLOGMESSAGE_INSERT

5.1.19. Table: IPCLOGMESSAGEHIS

Columns

	Name	Data type	Description / Attributes
	ID	numeric(18, 0)	
	IPCTRANSDATE	datetime	Default: getdate()
	IPCWORKDATE	datetime	
	IPCTRANSID	numeric(18, 0)	
	IPCLOGTYPE	char(2)	
	MESSAGETYPE	varchar(10)	
	MESSAGE	nvarchar(4000)	

TRIAL

5.1.20. Table: IPCLOGTRANAPPDETAIL

Columns

	Name	Data type	Description / Attributes
>List	IPCTRANSID	numeric(18, 0)	
List	IPCWORKDATE	datetime	
List	IPCTRANSDATE	datetime	
List	CONTENTVALUE	nvarchar(MAX)	
List	STATUS	varchar(1)	

Unique keys

	Columns	Name / Description
Key	IPCTRANSID	PK_IPCLOGTRANAPPDETAIL

Used By

	Name
Table	IPCLOGTRANAPPDETAIL
StoredProcedure	IPCLOGTRANAPPDETAIL_INSERT

5.1.21. Table: IPCLOGTRANAPPDETAILHIS

Columns

	Name	Data type	Description / Attributes
>List	IPCTRANSID	numeric(18, 0)	
Key	IPCWORKDATE	datetime	
List	IPCTRANSDATE	datetime	
List	CONTENTVALUE	nvarchar(MAX)	
List	STATUS	varchar(1)	

Unique keys

	Columns	Name / Description
Key	IPCTRANSID, IPCWORKDATE	PK_IPCLOGTRANAPPDETAILHIS

TRIAL

5.1.22. Table: IPCLOGTRANAPPROVE

Columns

	Name	Data type	Description / Attributes
>List	IPCTRANSID	numeric(18, 0)	
List	IPCTRANSDATE	datetime	
List	IPCWORKDATE	datetime	
List	USERAPP	varchar(50)	
List	AUTHENTYPE	varchar(50)	Nullable
List	AUTHENCODE	varchar(50)	Nullable

Unique keys

	Columns	Name / Description
Key	IPCTRANSID, USERAPP	PK_IPCLOGTRANAPPROVE

Used By

	Name
Table	IPCLOGTRANAPPROVE
Table	IPCAPPROVETRAN_TELLER
Table	IPCAPPROVETRAN_USER

5.1.23. Table: IPCLOGTRANAPPROVEHIS

Columns

	Name	Data type	Description / Attributes
>List	IPCTRANSID	numeric(18, 0)	
List	IPCTRANSDATE	datetime	
List	IPCWORKDATE	datetime	
List	USERAPP	varchar(50)	
List	AUTHENTYPE	varchar(50)	Nullable
List	AUTHENCODE	varchar(50)	Nullable

Unique keys

	Columns	Name / Description
Key	IPCTRANSID, IPCWORKDATE, USERAPP	PK_IPCLOGTRANAPPROVEHIS_1

TRIAL

5.1.24. Table: IPCLOGTRANS

Columns

	Name	Data type	Description / Attributes
	IPCTRANSDATE	datetime	Default: getdate()
	IPCWORKDATE	datetime	
	IPCTRANSID	numeric(18, 0)	
	IPCTRANCODE	varchar(20)	
	TRANREF	varchar(50)	Nullable
	SENDERBANK	varchar(20)	Nullable
	RECEIVERBANK	varchar(20)	Nullable
	SENDERCHANNEL	varchar(20)	Nullable
	RECEIVERCHANNEL	varchar(20)	Nullable
	CCYID	varchar(10)	Nullable
	SOURCEID	varchar(20)	
	DESTID	varchar(20)	
	SOURCETRANREF	varchar(20)	
	USERID	varchar(50)	
	USERCURAPP	varchar(50)	Nullable
	NEXTUSERAPP	varchar(50)	Nullable
	LISTUSERAPP	varchar(MAX)	Nullable
	TRANDESC	nvarchar(400)	
	STATUS	varchar(10)	
	STATUSHIS	varchar(100)	Nullable
	LISTUSERPRS	varchar(1000)	Nullable
	APPRSTS	int	
	OFFLSTS	varchar(1)	
	DELETED	bit	
	DESTTRANREF	varchar(20)	Nullable
	DESTERRORCODE	varchar(50)	Nullable
	ERRORCODE	varchar(50)	Default: ''
	ERRORDESC	nvarchar(1000)	Default: ''
	ONLINE	bit	Default: 1
	CHAR01	nvarchar(1000)	Nullable
	CHAR02	nvarchar(1000)	Nullable
	CHAR03	nvarchar(1000)	Nullable
	CHAR04	nvarchar(1000)	Nullable
	CHAR05	nvarchar(1000)	Nullable
	CHAR06	nvarchar(1000)	Nullable
	CHAR07	nvarchar(1000)	Nullable
	CHAR08	nvarchar(1000)	Nullable

	Name	Data type	Description / Attributes
█	CHAR09	nvarchar(1000)	Nullable
█	CHAR10	nvarchar(1000)	Nullable
█	CHAR11	nvarchar(1000)	Nullable
█	CHAR12	nvarchar(1000)	Nullable
█	CHAR13	nvarchar(1000)	Nullable
█	CHAR14	nvarchar(1000)	Nullable
█	CHAR15	nvarchar(1000)	Nullable
█	CHAR16	nvarchar(1000)	Nullable
█	CHAR17	nvarchar(1000)	Nullable
█	CHAR18	nvarchar(1000)	Nullable
█	CHAR19	nvarchar(1000)	Nullable
█	CHAR20	nvarchar(1000)	Nullable
█	CHAR21	nvarchar(1000)	Nullable
█	CHAR22	nvarchar(1000)	Nullable
█	CHAR23	nvarchar(1000)	Nullable
█	CHAR24	nvarchar(1000)	Nullable
█	CHAR25	nvarchar(1000)	Nullable
█	CHAR26	nvarchar(1000)	Nullable
█	CHAR27	nvarchar(1000)	Nullable
█	CHAR28	nvarchar(1000)	Nullable
█	CHAR29	nvarchar(1000)	Nullable
█	CHAR30	nvarchar(1000)	Nullable
█	CHAR31	nvarchar(1000)	Nullable
█	CHAR32	nvarchar(1000)	Nullable
█	CHAR33	nvarchar(1000)	Nullable
█	CHAR34	nvarchar(1000)	Nullable
█	CHAR35	nvarchar(1000)	Nullable
█	CHAR36	nvarchar(1000)	Nullable
█	CHAR37	nvarchar(1000)	Nullable
█	CHAR38	nvarchar(1000)	Nullable
█	CHAR39	nvarchar(1000)	Nullable
█	CHAR40	nvarchar(1000)	Nullable
█	CHAR41	nvarchar(1000)	Nullable
█	CHAR42	nvarchar(1000)	Nullable
█	CHAR43	nvarchar(1000)	Nullable
█	CHAR44	nvarchar(1000)	Nullable
█	CHAR45	nvarchar(1000)	Nullable
█	CHAR46	nvarchar(1000)	Nullable
█	CHAR47	nvarchar(1000)	Nullable
█	CHAR48	nvarchar(1000)	Nullable

	Name	Data type	Description / Attributes
1	CHAR49	nvarchar(1000)	Nullable
2	CHAR50	nvarchar(1000)	Nullable
3	CHAR51	nvarchar(1000)	Nullable
4	CHAR52	nvarchar(1000)	Nullable
5	CHAR53	nvarchar(1000)	Nullable
6	CHAR54	nvarchar(1000)	Nullable
7	CHAR55	nvarchar(1000)	Nullable
8	CHAR56	nvarchar(1000)	Nullable
9	CHAR57	nvarchar(1000)	Nullable
10	CHAR58	nvarchar(1000)	Nullable
11	CHAR59	nvarchar(1000)	Nullable
12	CHAR60	nvarchar(1000)	Nullable
13	CHAR61	nvarchar(1000)	Nullable
14	CHAR62	nvarchar(1000)	Nullable
15	CHAR63	nvarchar(1000)	Nullable
16	CHAR64	nvarchar(1000)	Nullable
17	CHAR65	nvarchar(1000)	Nullable
18	CHAR66	nvarchar(1000)	Nullable
19	CHAR67	nvarchar(1000)	Nullable
20	CHAR68	nvarchar(1000)	Nullable
21	CHAR69	nvarchar(1000)	Nullable
22	CHAR70	nvarchar(1000)	Nullable
23	NUM01	numeric(20, 4)	Nullable
24	NUM02	numeric(20, 4)	Nullable
25	NUM03	numeric(20, 4)	Nullable
26	NUM04	numeric(20, 4)	Nullable
27	NUM05	numeric(20, 4)	Nullable
28	NUM06	numeric(20, 4)	Nullable
29	NUM07	numeric(20, 4)	Nullable
30	NUM08	numeric(20, 4)	Nullable
31	NUM09	numeric(20, 4)	Nullable
32	NUM10	numeric(20, 4)	Nullable
33	NUM11	numeric(20, 4)	Nullable
34	NUM12	numeric(20, 4)	Nullable
35	NUM13	numeric(20, 4)	Nullable
36	NUM14	numeric(20, 4)	Nullable
37	NUM15	numeric(20, 4)	Nullable
38	NUM16	numeric(20, 4)	Nullable
39	NUM17	numeric(20, 4)	Nullable
40	NUM18	numeric(20, 4)	Nullable

	Name	Data type	Description / Attributes
1	NUM19	numeric(20, 4)	Nullable
2	NUM20	numeric(20, 4)	Nullable
3	NUM21	numeric(20, 4)	Nullable
4	NUM22	numeric(20, 4)	Nullable
5	NUM23	numeric(20, 4)	Nullable
6	NUM24	numeric(20, 4)	Nullable
7	NUM25	numeric(20, 4)	Nullable
8	NUM26	numeric(20, 4)	Nullable
9	NUM27	numeric(20, 4)	Nullable
10	NUM28	numeric(20, 4)	Nullable
11	NUM29	numeric(20, 4)	Nullable
12	NUM30	numeric(20, 4)	Nullable
13	NUM31	numeric(20, 4)	Nullable
14	NUM32	numeric(20, 4)	Nullable
15	NUM33	numeric(20, 4)	Nullable
16	NUM34	numeric(20, 4)	Nullable
17	NUM35	numeric(20, 4)	Nullable
18	NUM36	numeric(20, 4)	Nullable
19	NUM37	numeric(20, 4)	Nullable
20	NUM38	numeric(20, 4)	Nullable
21	NUM39	numeric(20, 4)	Nullable
22	NUM40	numeric(20, 4)	Nullable
23	NUM41	numeric(20, 4)	Nullable
24	NUM42	numeric(20, 4)	Nullable
25	NUM43	numeric(20, 4)	Nullable
26	NUM44	numeric(20, 4)	Nullable
27	NUM45	numeric(20, 4)	Nullable
28	NUM46	numeric(20, 4)	Nullable
29	NUM47	numeric(20, 4)	Nullable
30	NUM48	numeric(20, 4)	Nullable
31	NUM49	numeric(20, 4)	Nullable
32	NUM50	numeric(20, 4)	Nullable
33	NUM51	numeric(20, 4)	Nullable
34	NUM52	numeric(20, 4)	Nullable
35	NUM53	numeric(20, 4)	Nullable
36	NUM54	numeric(20, 4)	Nullable
37	NUM55	numeric(20, 4)	Nullable
38	NUM56	numeric(20, 4)	Nullable
39	NUM57	numeric(20, 4)	Nullable
40	NUM58	numeric(20, 4)	Nullable

	Name	Data type	Description / Attributes
█	NUM59	numeric(20, 4)	Nullable
█	NUM60	numeric(20, 4)	Nullable
█	NUM61	numeric(20, 4)	Nullable
█	NUM62	numeric(20, 4)	Nullable
█	NUM63	numeric(20, 4)	Nullable
█	NUM64	numeric(20, 4)	Nullable
█	NUM65	numeric(20, 4)	Nullable
█	NUM66	numeric(20, 4)	Nullable
█	NUM67	numeric(20, 4)	Nullable
█	NUM68	numeric(20, 4)	Nullable
█	NUM69	numeric(20, 4)	Nullable
█	NUM70	numeric(20, 4)	Nullable
█	ISBATCH	varchar(1)	Nullable
█	BATCHREF	varchar(50)	Nullable

Unique keys

	Columns	Name / Description
█	IPCWORKDATE, IPCTRANSID	PK_IPCLOGTRANS_1

Used By

	Name
IPCLOGTRANS	
⚙ FAST_GETOUTGOING (FAST GET Outgoing Transaction)	
⚙ FAST_GETTRANBYFCNTXREFID (Get Transaction By FCN)	
⚙ FAST_GETTRANBYPMTINFID (Get Transaction Payment ID)	
⚙ FAST_GETTRANSDetail (FAST Get transaction detail)	
⚙ FAST_RESEND_TX	
⚙ FAST_SELECTTRANBYPAYMENTID (FAST Select transaction payment ID)	
⚙ IPC_GETDESTTRANREF	
⚙ IPC_UPDATEBATCHTRANS	
⚙ IPC_UPDATELISTREVERSAL	
⚙ IPC_UPDATELOGDEFINE	
⚙ IPC_UPDATELOGTRANSErrorROR	
⚙ IPCAPPROVETRAN_TELLER	
⚙ IPCAPPROVETRAN_USER	
⚙ IPCCHECKEXIST_TRANREF	
⚙ IPCLOGTRANS_GETTRANSACTION_TOFAST (Get transaction send to FAST)	
⚙ IPCLOGTRANS_INSERT	
⚙ IPCLOGTRANS_OFFLINE_SELECT	
⚙ IPCLOGTRANS_OFFLINE_UPDATE	
⚙ IPCLOGTRANS_REVERSAL	

Name
IPCLOGTRANS_SYNCO9_UPDATE
IPCLOGTRANS_TIMEOUT
IPCLOGTRANS_UPDATE
IPCLOGTRANS_UPDATE_REVERSE_REFERENCE
IPCLOGTRANS_UPDATEAPPROVE
IPCLOGTRANS_UPDATEFATHERTRANS
IPCLOGTRANS_UPDATEREAD (Log transaction Update Status)
IPCLOGTRANS_UPDATEREAD_SYNC TO CORE (Log transaction update status sync to core)
IPCLOGTRANS_UPDATETIMEOUT

TRIAL

5.1.25. Table: IPCLOGTRANSDetail

Columns

		Name	Data type	Description / Attributes
		IPCTRANSDATE	datetime	Default: getdate()
		IPCWORKDATE	datetime	
		IPCTRANSID	numeric(18, 0)	
		LOGTYPE	varchar(1)	
		FIELDNAME	varchar(50)	
		FIELDVALUE	nvarchar(2000)	

Unique keys

Columns		Name / Description
	IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME	PK_IPCLOGTRANSDetail

Used By

Name	
	IPCLOGTRANSDetail
	IPCLOGTRANSDetail_INSERT
	IPCLOGTRANSDetail_SELECT

5.1.26. Table: IPCLOGTRANSDETAILFIELD

Columns

	Name	Data type	Description / Attributes
█	SOURCEID	varchar(20)	
█	DESTID	varchar(20)	
█	IPCTRANCODE	varchar(20)	
█	FIELDNAME	varchar(50)	
█	VALUESTYLEON	varchar(10)	
█	VALUENAMEON	varchar(50)	
█	VALUESTYLEOFF	varchar(10)	
█	VALUENAMEOFF	varchar(50)	
█	LOGTYPE	varchar(1)	

Unique keys

	Columns	Name / Description
█	SOURCEID, DESTID, IPCTRANCODE, FIELDNAME, LOGTYPE	PK_IPCLOGTRANSDETAILFIELD

Used By

	Name
█	IPCLOGTRANSDETAILFIELD
⚙	IPCLOGTRANSDETAILFIELD_SELECT

5.1.27. Table: IPCLOGTRANSDetailHIS

Columns

	Name	Data type	Description / Attributes
	IPCTRANSDATE	datetime	
	IPCWORKDATE	datetime	
	IPCTRANSID	numeric(18, 0)	
	LOGTYPE	varchar(1)	
	FIELDNAME	varchar(50)	
	FIELDVALUE	nvarchar(2000)	

Unique keys

	Columns	Name / Description
	IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME	PK_IPCLOGTRANSDetailHIS

TRIAL

5.1.28. Table: IPCLOGTRANSEVENT

Columns

	Name	Data type	Description / Attributes
	IPCTRANSDATE	datetime	Default: getdate()
	IPCWORKDATE	datetime	
	IPCTRANSID	numeric(18, 0)	
	IPCTRANCODE	varchar(20)	
	CCYID	varchar(10)	Nullable
	SOURCEID	varchar(20)	
	SOURCETRANREF	varchar(20)	
	USERID	varchar(50)	
	USERCURAPP	varchar(50)	Nullable
	NEXTUSERAPP	varchar(50)	Nullable
	LISTUSERAPP	varchar(MAX)	Nullable
	TRANDESC	nvarchar(400)	
	STATUS	varchar(10)	
	STATUSHIS	varchar(100)	Nullable
	LISTUSERPRS	varchar(1000)	Nullable
	APPRSTS	int	
	OFFLSTS	varchar(1)	
	DELETED	bit	
	DESTID	varchar(20)	
	DESTTRANREF	varchar(20)	Nullable
	DESTERRORCODE	varchar(50)	Nullable
	ERRORCODE	varchar(50)	Default: ''
	ERRORDESC	nvarchar(1000)	Default: ''
	ONLINE	bit	Default: 1
	CHAR01	nvarchar(1000)	Nullable
	CHAR02	nvarchar(1000)	Nullable
	CHAR03	nvarchar(1000)	Nullable
	CHAR04	nvarchar(1000)	Nullable
	CHAR05	nvarchar(1000)	Nullable
	CHAR06	nvarchar(1000)	Nullable
	CHAR07	nvarchar(1000)	Nullable
	CHAR08	nvarchar(1000)	Nullable
	CHAR09	nvarchar(1000)	Nullable
	CHAR10	nvarchar(1000)	Nullable
	CHAR11	nvarchar(1000)	Nullable
	CHAR12	nvarchar(1000)	Nullable
	CHAR13	nvarchar(1000)	Nullable

	Name	Data type	Description / Attributes
█	CHAR14	nvarchar(1000)	Nullable
█	CHAR15	nvarchar(1000)	Nullable
█	CHAR16	nvarchar(1000)	Nullable
█	CHAR17	nvarchar(1000)	Nullable
█	CHAR18	nvarchar(1000)	Nullable
█	CHAR19	nvarchar(1000)	Nullable
█	CHAR20	nvarchar(1000)	Nullable
█	CHAR21	nvarchar(1000)	Nullable
█	CHAR22	nvarchar(1000)	Nullable
█	CHAR23	nvarchar(1000)	Nullable
█	CHAR24	nvarchar(1000)	Nullable
█	CHAR25	nvarchar(1000)	Nullable
█	CHAR26	nvarchar(1000)	Nullable
█	CHAR27	nvarchar(1000)	Nullable
█	CHAR28	nvarchar(1000)	Nullable
█	CHAR29	nvarchar(1000)	Nullable
█	CHAR30	nvarchar(1000)	Nullable
█	CHAR31	nvarchar(1000)	Nullable
█	CHAR32	nvarchar(1000)	Nullable
█	CHAR33	nvarchar(1000)	Nullable
█	CHAR34	nvarchar(1000)	Nullable
█	CHAR35	nvarchar(1000)	Nullable
█	CHAR36	nvarchar(1000)	Nullable
█	CHAR37	nvarchar(1000)	Nullable
█	CHAR38	nvarchar(1000)	Nullable
█	CHAR39	nvarchar(1000)	Nullable
█	CHAR40	nvarchar(1000)	Nullable
█	CHAR41	nvarchar(1000)	Nullable
█	CHAR42	nvarchar(1000)	Nullable
█	CHAR43	nvarchar(1000)	Nullable
█	CHAR44	nvarchar(1000)	Nullable
█	CHAR45	nvarchar(1000)	Nullable
█	CHAR46	nvarchar(1000)	Nullable
█	CHAR47	nvarchar(1000)	Nullable
█	CHAR48	nvarchar(1000)	Nullable
█	CHAR49	nvarchar(1000)	Nullable
█	CHAR50	nvarchar(1000)	Nullable
█	CHAR51	nvarchar(1000)	Nullable
█	CHAR52	nvarchar(1000)	Nullable
█	CHAR53	nvarchar(1000)	Nullable

	Name	Data type	Description / Attributes
█	CHAR54	nvarchar(1000)	Nullable
█	CHAR55	nvarchar(1000)	Nullable
█	CHAR56	nvarchar(1000)	Nullable
█	CHAR57	nvarchar(1000)	Nullable
█	CHAR58	nvarchar(1000)	Nullable
█	CHAR59	nvarchar(1000)	Nullable
█	CHAR60	nvarchar(1000)	Nullable
█	CHAR61	nvarchar(1000)	Nullable
█	CHAR62	nvarchar(1000)	Nullable
█	CHAR63	nvarchar(1000)	Nullable
█	CHAR64	nvarchar(1000)	Nullable
█	CHAR65	nvarchar(1000)	Nullable
█	CHAR66	nvarchar(1000)	Nullable
█	CHAR67	nvarchar(1000)	Nullable
█	CHAR68	nvarchar(1000)	Nullable
█	CHAR69	nvarchar(1000)	Nullable
█	CHAR70	nvarchar(1000)	Nullable
█	NUM01	numeric(20, 4)	Nullable
█	NUM02	numeric(20, 4)	Nullable
█	NUM03	numeric(20, 4)	Nullable
█	NUM04	numeric(20, 4)	Nullable
█	NUM05	numeric(20, 4)	Nullable
█	NUM06	numeric(20, 4)	Nullable
█	NUM07	numeric(20, 4)	Nullable
█	NUM08	numeric(20, 4)	Nullable
█	NUM09	numeric(20, 4)	Nullable
█	NUM10	numeric(20, 4)	Nullable
█	NUM11	numeric(20, 4)	Nullable
█	NUM12	numeric(20, 4)	Nullable
█	NUM13	numeric(20, 4)	Nullable
█	NUM14	numeric(20, 4)	Nullable
█	NUM15	numeric(20, 4)	Nullable
█	NUM16	numeric(20, 4)	Nullable
█	NUM17	numeric(20, 4)	Nullable
█	NUM18	numeric(20, 4)	Nullable
█	NUM19	numeric(20, 4)	Nullable
█	NUM20	numeric(20, 4)	Nullable
█	NUM21	numeric(20, 4)	Nullable
█	NUM22	numeric(20, 4)	Nullable
█	NUM23	numeric(20, 4)	Nullable

	Name	Data type	Description / Attributes
1	NUM24	numeric(20, 4)	Nullable
2	NUM25	numeric(20, 4)	Nullable
3	NUM26	numeric(20, 4)	Nullable
4	NUM27	numeric(20, 4)	Nullable
5	NUM28	numeric(20, 4)	Nullable
6	NUM29	numeric(20, 4)	Nullable
7	NUM30	numeric(20, 4)	Nullable
8	NUM31	numeric(20, 4)	Nullable
9	NUM32	numeric(20, 4)	Nullable
10	NUM33	numeric(20, 4)	Nullable
11	NUM34	numeric(20, 4)	Nullable
12	NUM35	numeric(20, 4)	Nullable
13	NUM36	numeric(20, 4)	Nullable
14	NUM37	numeric(20, 4)	Nullable
15	NUM38	numeric(20, 4)	Nullable
16	NUM39	numeric(20, 4)	Nullable
17	NUM40	numeric(20, 4)	Nullable
18	NUM41	numeric(20, 4)	Nullable
19	NUM42	numeric(20, 4)	Nullable
20	NUM43	numeric(20, 4)	Nullable
21	NUM44	numeric(20, 4)	Nullable
22	NUM45	numeric(20, 4)	Nullable
23	NUM46	numeric(20, 4)	Nullable
24	NUM47	numeric(20, 4)	Nullable
25	NUM48	numeric(20, 4)	Nullable
26	NUM49	numeric(20, 4)	Nullable
27	NUM50	numeric(20, 4)	Nullable
28	NUM51	numeric(20, 4)	Nullable
29	NUM52	numeric(20, 4)	Nullable
30	NUM53	numeric(20, 4)	Nullable
31	NUM54	numeric(20, 4)	Nullable
32	NUM55	numeric(20, 4)	Nullable
33	NUM56	numeric(20, 4)	Nullable
34	NUM57	numeric(20, 4)	Nullable
35	NUM58	numeric(20, 4)	Nullable
36	NUM59	numeric(20, 4)	Nullable
37	NUM60	numeric(20, 4)	Nullable
38	NUM61	numeric(20, 4)	Nullable
39	NUM62	numeric(20, 4)	Nullable
40	NUM63	numeric(20, 4)	Nullable

	Name	Data type	Description / Attributes
█	NUM64	numeric(20, 4)	Nullable
█	NUM65	numeric(20, 4)	Nullable
█	NUM66	numeric(20, 4)	Nullable
█	NUM67	numeric(20, 4)	Nullable
█	NUM68	numeric(20, 4)	Nullable
█	NUM69	numeric(20, 4)	Nullable
█	NUM70	numeric(20, 4)	Nullable
█	ISBATCH	varchar(1)	Nullable
█	BATCHREF	varchar(50)	Nullable
█	PARENTREF	numeric(18, 0)	Nullable

Unique keys

	Columns	Name / Description
█	IPCWORKDATE, IPCTRANSID	PK_IPCLOGTRANS1

Used By

	Name
█	IPCLOGTRANSEVENT
⚙	IPC_UPDATELOGDEFINE_EVENT (IPC Update Log Define Event)
⚙	IPCLOGTRANS_INSERT_EVENT
⚙	IPCLOGTRANS_UPDATE_EVENT
⚙	IPCLOGTRANS_UPDATEFATHERTRANS

5.1.29. Table: IPCLOGTRANSHIS

Columns

	Name	Data type	Description / Attributes
	IPCTRANSDATE	datetime	Default: getdate()
	IPCWORKDATE	datetime	
	IPCTRANSID	numeric(18, 0)	
	IPCTRANCODE	varchar(20)	
	CCYID	varchar(10)	Nullable
	SOURCEID	varchar(20)	
	SOURCETRANREF	varchar(20)	
	USERID	varchar(50)	
	USERCURAPP	varchar(50)	Nullable
	NEXTUSERAPP	varchar(50)	Nullable
	LISTUSERAPP	varchar(MAX)	Nullable
	TRANDESC	nvarchar(400)	
	STATUS	varchar(10)	
	STATUSHIS	varchar(100)	Nullable
	LISTUSERPRS	varchar(1000)	Nullable
	APPRSTS	int	
	OFFLSTS	varchar(1)	
	DELETED	bit	
	DESTID	varchar(20)	
	DESTTRANREF	varchar(20)	Nullable
	DESTERRORCODE	varchar(50)	Nullable
	ERRORCODE	varchar(50)	Default: ''
	ERRORDESC	nvarchar(1000)	Default: ''
	ONLINE	bit	Default: 1
	CHAR01	nvarchar(1000)	Nullable
	CHAR02	nvarchar(1000)	Nullable
	CHAR03	nvarchar(1000)	Nullable
	CHAR04	nvarchar(1000)	Nullable
	CHAR05	nvarchar(1000)	Nullable
	CHAR06	nvarchar(1000)	Nullable
	CHAR07	nvarchar(1000)	Nullable
	CHAR08	nvarchar(1000)	Nullable
	CHAR09	nvarchar(1000)	Nullable
	CHAR10	nvarchar(1000)	Nullable
	CHAR11	nvarchar(1000)	Nullable
	CHAR12	nvarchar(1000)	Nullable
	CHAR13	nvarchar(1000)	Nullable

	Name	Data type	Description / Attributes
█	CHAR14	nvarchar(1000)	Nullable
█	CHAR15	nvarchar(1000)	Nullable
█	CHAR16	nvarchar(1000)	Nullable
█	CHAR17	nvarchar(1000)	Nullable
█	CHAR18	nvarchar(1000)	Nullable
█	CHAR19	nvarchar(1000)	Nullable
█	CHAR20	nvarchar(1000)	Nullable
█	CHAR21	nvarchar(1000)	Nullable
█	CHAR22	nvarchar(1000)	Nullable
█	CHAR23	nvarchar(1000)	Nullable
█	CHAR24	nvarchar(1000)	Nullable
█	CHAR25	nvarchar(1000)	Nullable
█	CHAR26	nvarchar(1000)	Nullable
█	CHAR27	nvarchar(1000)	Nullable
█	CHAR28	nvarchar(1000)	Nullable
█	CHAR29	nvarchar(1000)	Nullable
█	CHAR30	nvarchar(1000)	Nullable
█	CHAR31	nvarchar(1000)	Nullable
█	CHAR32	nvarchar(1000)	Nullable
█	CHAR33	nvarchar(1000)	Nullable
█	CHAR34	nvarchar(1000)	Nullable
█	CHAR35	nvarchar(1000)	Nullable
█	CHAR36	nvarchar(1000)	Nullable
█	CHAR37	nvarchar(1000)	Nullable
█	CHAR38	nvarchar(1000)	Nullable
█	CHAR39	nvarchar(1000)	Nullable
█	CHAR40	nvarchar(1000)	Nullable
█	CHAR41	nvarchar(1000)	Nullable
█	CHAR42	nvarchar(1000)	Nullable
█	CHAR43	nvarchar(1000)	Nullable
█	CHAR44	nvarchar(1000)	Nullable
█	CHAR45	nvarchar(1000)	Nullable
█	CHAR46	nvarchar(1000)	Nullable
█	CHAR47	nvarchar(1000)	Nullable
█	CHAR48	nvarchar(1000)	Nullable
█	CHAR49	nvarchar(1000)	Nullable
█	CHAR50	nvarchar(1000)	Nullable
█	CHAR51	nvarchar(1000)	Nullable
█	CHAR52	nvarchar(1000)	Nullable
█	CHAR53	nvarchar(1000)	Nullable

	Name	Data type	Description / Attributes
█	CHAR54	nvarchar(1000)	Nullable
█	CHAR55	nvarchar(1000)	Nullable
█	CHAR56	nvarchar(1000)	Nullable
█	CHAR57	nvarchar(1000)	Nullable
█	CHAR58	nvarchar(1000)	Nullable
█	CHAR59	nvarchar(1000)	Nullable
█	CHAR60	nvarchar(1000)	Nullable
█	CHAR61	nvarchar(1000)	Nullable
█	CHAR62	nvarchar(1000)	Nullable
█	CHAR63	nvarchar(1000)	Nullable
█	CHAR64	nvarchar(1000)	Nullable
█	CHAR65	nvarchar(1000)	Nullable
█	CHAR66	nvarchar(1000)	Nullable
█	CHAR67	nvarchar(1000)	Nullable
█	CHAR68	nvarchar(1000)	Nullable
█	CHAR69	nvarchar(1000)	Nullable
█	CHAR70	nvarchar(1000)	Nullable
█	NUM01	numeric(20, 4)	Nullable
█	NUM02	numeric(20, 4)	Nullable
█	NUM03	numeric(20, 4)	Nullable
█	NUM04	numeric(20, 4)	Nullable
█	NUM05	numeric(20, 4)	Nullable
█	NUM06	numeric(20, 4)	Nullable
█	NUM07	numeric(20, 4)	Nullable
█	NUM08	numeric(20, 4)	Nullable
█	NUM09	numeric(20, 4)	Nullable
█	NUM10	numeric(20, 4)	Nullable
█	NUM11	numeric(20, 4)	Nullable
█	NUM12	numeric(20, 4)	Nullable
█	NUM13	numeric(20, 4)	Nullable
█	NUM14	numeric(20, 4)	Nullable
█	NUM15	numeric(20, 4)	Nullable
█	NUM16	numeric(20, 4)	Nullable
█	NUM17	numeric(20, 4)	Nullable
█	NUM18	numeric(20, 4)	Nullable
█	NUM19	numeric(20, 4)	Nullable
█	NUM20	numeric(20, 4)	Nullable
█	NUM21	numeric(20, 4)	Nullable
█	NUM22	numeric(20, 4)	Nullable
█	NUM23	numeric(20, 4)	Nullable

	Name	Data type	Description / Attributes
1	NUM24	numeric(20, 4)	Nullable
2	NUM25	numeric(20, 4)	Nullable
3	NUM26	numeric(20, 4)	Nullable
4	NUM27	numeric(20, 4)	Nullable
5	NUM28	numeric(20, 4)	Nullable
6	NUM29	numeric(20, 4)	Nullable
7	NUM30	numeric(20, 4)	Nullable
8	NUM31	numeric(20, 4)	Nullable
9	NUM32	numeric(20, 4)	Nullable
10	NUM33	numeric(20, 4)	Nullable
11	NUM34	numeric(20, 4)	Nullable
12	NUM35	numeric(20, 4)	Nullable
13	NUM36	numeric(20, 4)	Nullable
14	NUM37	numeric(20, 4)	Nullable
15	NUM38	numeric(20, 4)	Nullable
16	NUM39	numeric(20, 4)	Nullable
17	NUM40	numeric(20, 4)	Nullable
18	NUM41	numeric(20, 4)	Nullable
19	NUM42	numeric(20, 4)	Nullable
20	NUM43	numeric(20, 4)	Nullable
21	NUM44	numeric(20, 4)	Nullable
22	NUM45	numeric(20, 4)	Nullable
23	NUM46	numeric(20, 4)	Nullable
24	NUM47	numeric(20, 4)	Nullable
25	NUM48	numeric(20, 4)	Nullable
26	NUM49	numeric(20, 4)	Nullable
27	NUM50	numeric(20, 4)	Nullable
28	NUM51	numeric(20, 4)	Nullable
29	NUM52	numeric(20, 4)	Nullable
30	NUM53	numeric(20, 4)	Nullable
31	NUM54	numeric(20, 4)	Nullable
32	NUM55	numeric(20, 4)	Nullable
33	NUM56	numeric(20, 4)	Nullable
34	NUM57	numeric(20, 4)	Nullable
35	NUM58	numeric(20, 4)	Nullable
36	NUM59	numeric(20, 4)	Nullable
37	NUM60	numeric(20, 4)	Nullable
38	NUM61	numeric(20, 4)	Nullable
39	NUM62	numeric(20, 4)	Nullable
40	NUM63	numeric(20, 4)	Nullable

	Name	Data type	Description / Attributes
1	NUM64	numeric(20, 4)	Nullable
2	NUM65	numeric(20, 4)	Nullable
3	NUM66	numeric(20, 4)	Nullable
4	NUM67	numeric(20, 4)	Nullable
5	NUM68	numeric(20, 4)	Nullable
6	NUM69	numeric(20, 4)	Nullable
7	NUM70	numeric(20, 4)	Nullable
8	ISBATCH	varchar(1)	Nullable
9	BATCHREF	varchar(50)	Nullable
10	PARENTREF	numeric(18, 0)	Nullable

Unique keys

	Columns	Name / Description
1	IPCWORKDATE, IPCTRANSID	PK_IPCLOGTRANSHIS

TRIAL

5.1.30. Table: IPCLOGTRANSREVERSAL

Columns

	Name	Data type	Description / Attributes
█	IPCTRANSID	numeric(18, 0)	
█	IPCTRANCODE	varchar(50)	
█	RVSTRANCODE	varchar(50)	
█	RVSDESTID	varchar(50)	
█	IPCWORKDATE	nvarchar(50)	Nullable
█	DATETIMEINSERT	datetime	
█	PRIORITY	int	Nullable
█	COUNT	int	Nullable Default: 0
█	ERRORCODE	nvarchar(200)	Nullable
█	ERRORDESC	nvarchar(200)	Nullable
█	DATA	nvarchar(MAX)	Nullable

Unique keys

	Columns	Name / Description
█	IPCTRANSID, RVSTRANCODE	PK_IPCLOGTRANSREVERSAL_1

Used By

	Name
█	IPCLOGTRANSREVERSAL
⚙	IPC_CHECKDESTTRANS
⚙	IPC_CHECKREVERSAL
⚙	IPC_GETLISTREVERSAL
⚙	IPC_UPDATELISTREVERSAL

5.1.31. Table: IPCLOGTRANSREVERSALHIS

Columns

	Name	Data type	Description / Attributes
1	IPCTRANSID	numeric(18, 0)	
2	IPCTRANCODE	varchar(50)	Nullable
3	RVSTRANCODE	varchar(50)	Nullable
4	RVSDESTID	varchar(50)	Nullable
5	IPCWORKDATE	nvarchar(50)	Nullable
6	DATETIMEINSERT	datetime	Nullable
7	PRIORITY	int	Nullable
8	COUNT	int	Nullable Default: 0
9	DATETIMEPROCESS	datetime	Nullable
10	ERRORCODE	nvarchar(200)	Nullable
11	ERRORDESC	nvarchar(200)	Nullable
12	DATA	nvarchar(MAX)	Nullable

Used By

	Name
1	IPCLOGTRANSREVERSALHIS
2	IPC_UPDATELISTREVERSAL

5.1.32. Table: IPCLOGTRANSREVERSALTELLER

Columns

	Name	Data type	Description / Attributes
>List	IPCTRANSID	numeric(18, 0)	
List	BRANCHID	varchar(50)	
List	IPCWORKDATE	varchar(50)	
List	STATUS	varchar(5)	
List	USERCREATE	varchar(50)	
List	DATECREATE	datetime	
List	USERAPPROVE	varchar(50)	Nullable
List	DATEAPPROVE	datetime	Nullable

Unique keys

	Columns	Name / Description
Key	IPCTRANSID	PK_IPCLOGTRANSREVERSALTELLER

5.1.33. Table: IPCMAPPINGCOLNAME

Columns

	Name	Data type	Description / Attributes
1	IPCTRANCODE	varchar(50)	
2	DESTID	varchar(50)	
3	DSNAME	varchar(50)	
4	TABLENAME	varchar(50)	
5	TARGETNAME	varchar(50)	
6	MAPPINGNAME	varchar(50)	
7	ORD	int	

Used By

Name
IPC MAPPING COLNAME
IPC GET MAPPING COLNAME

5.1.34. Table: IPCOUTPUTDEFINEHAS

Columns

	Name	Data type	Description / Attributes
█	SOURCEID	varchar(20)	
█	DESTID	varchar(20)	
█	IPCTRANCODE	varchar(20)	
█	ONLINE	bit	
█	FIELDNO	int	
█	FIELDDESC	varchar(100)	
█	FIELDSTYLE	varchar(50)	
█	FIELDNAME	nvarchar(1000)	
█	VALUESTYLE	varchar(50)	
█	VALUENAME	nvarchar(1000)	
█	FORMATTYPE	varchar(10)	
█	FORMATOBJECT	varchar(1000)	
█	FORMATFUNCTION	varchar(1000)	
█	FORMATPARM	varchar(1000)	

Unique keys

	Columns	Name / Description
█	SOURCEID, DESTID, IPCTRANCODE, ONLINE, FIELDNO	PK_IPCOUTPUTDEFINEHAS

5.1.35. Table: IPCOUTPUTDEFINEHTTPHEADER

Columns

	Name	Data type	Description / Attributes
█	SOURCEID	varchar(20)	
█	DESTID	varchar(20)	
█	IPCTRANCODE	varchar(20)	
█	ONLINE	bit	
█	FIELDNO	int	
█	FIELDDESC	varchar(100)	
█	FIELDSTYLE	varchar(50)	
█	FIELDNAME	nvarchar(1000)	
█	VALUESTYLE	varchar(10)	
█	VALUENAME	nvarchar(1000)	
█	FORMATTYPE	varchar(10)	
█	FORMATOBJECT	varchar(1000)	
█	FORMATFUNCTION	varchar(1000)	
█	FORMATPARM	varchar(1000)	

Unique keys

	Columns	Name / Description
█	SOURCEID, DESTID, IPCTRANCODE, ONLINE, FIELDNO	PK_IPCOUTPUTDEFINEHTTPHEADER

5.1.36. Table: IPCOUTPUTDEFINEXML

Columns

	Name	Data type	Description / Attributes
█	SOURCEID	varchar(20)	
█	DESTID	varchar(20)	
█	IPCTRANCODE	varchar(20)	
█	ONLINE	bit	
█	FIELDNO	int	
█	FIELDDESC	varchar(100)	
█	FIELDSTYLE	varchar(50)	
█	FIELDNAME	nvarchar(1000)	
█	VALUESTYLE	varchar(10)	
█	VALUENAME	nvarchar(1000)	
█	FORMATTYPE	varchar(10)	
█	FORMATOBJECT	varchar(1000)	
█	FORMATFUNCTION	varchar(1000)	
█	FORMATPARM	varchar(1000)	
█	ARRAYLEVEL	int	Default: 0
█	SUBLEVEL	int	Default: 0
█	ARRAYNODE	varchar(100)	Nullable

Unique keys

	Columns	Name / Description
█	SOURCEID, DESTID, IPCTRANCODE, ONLINE, FIELDNO	PK_IPCOUTPUTDEFINEXML

5.1.37. Table: IPCOUTPUTMODIFYHTTP

Columns

	Name	Data type	Description / Attributes
█	SOURCEID	varchar(20)	
█	DESTID	varchar(20)	
█	IPCTRANCODE	varchar(20)	
█	ONLINE	bit	
█	FIELDNO	int	
█	FIELDDESC	varchar(100)	
█	FIELDSTYLE	varchar(50)	
█	FIELDNAME	nvarchar(1000)	
█	VALUESTYLE	varchar(10)	
█	VALUENAME	nvarchar(1000)	
█	FORMATTYPE	varchar(10)	
█	FORMATOBJECT	varchar(1000)	
█	FORMATFUNCTION	varchar(1000)	
█	FORMATPARM	varchar(1000)	

Unique keys

	Columns	Name / Description
█	SOURCEID, DESTID, IPCTRANCODE, ONLINE, FIELDNO	PK_IPCOUTPUTMODIFYHTTP

5.1.38. Table: IPCPARMINFO

Columns

	Name	Data type	Description / Attributes
█	SOURCEID	varchar(20)	
█	DESTID	varchar(20)	
█	IPCTRANCODE	varchar(20)	
█	PARENTNODE	varchar(20)	
█	NODE	varchar(20)	
█	MAPPINGSTYLE	varchar(10)	
█	MAPPINGVALUE	nvarchar(200)	
█	POSITION	smallint	
█	DESCRIPTION	nvarchar(200)	

Unique keys

	Columns	Name / Description
█	SOURCEID, DESTID, IPCTRANCODE, PARENTNODE, NODE	PK_IPCPARMINFO

5.1.39. Table: IPCSERVICE

Columns

Name		Data type	Description / Attributes
	ServiceID	varchar(50)	
	ServiceName	nvarchar(255)	
	Status	varchar(1)	
	BankService	bit	Nullable
	CorpService	bit	Nullable
	checkuseronline	bit	Nullable

Unique keys

Columns		Name / Description
	ServiceID	PK_IPC_SERVICE

TRIAL

5.1.40. Table: IPCSYSVAR

Columns

	Name	Data type	Description / Attributes
	VARNAME	varchar(50)	
	VARVALUE	nvarchar(4000)	
	VARDESC	nvarchar(1000)	Nullable

Unique keys

	Columns	Name / Description
	VARNAME	PK_IPCSYSVAR

Used By

	Name
	IPCSYSVAR
	FAST resend_tx
	IPC_CHECKDESTTRANS
	IPC_CHECKREVERSAL
	IPC_GETLISTREVERSAL
	IPC_GETTRANSID_TEMP
	IPC_GetWorkingDate
	IPC_UPDATE_SESSION
	IPCCOMPONENT_SELECT
	IPCLOGTRANS_GETTRANSSYNC09
	IPCLOGTRANS_GETTRANSSYNCREFUND
	IPCSYSVAR_INSERT09SESSION
	IPCSYSVAR_SELECT
	IPCSYSVAR_UPDATE

5.1.41. Table: IPCTRANSLIST

Columns

	Name	Data type	Description / Attributes
█	IPCTRANCODE	varchar(20)	
█	TRANDESC	nvarchar(200)	
█	SHORTTRANDESC	nvarchar(200)	Nullable
█	SERVICEID	varchar(10)	
█	ACCTYPE	varchar(10)	Nullable
█	TRANTYPE	varchar(50)	Nullable
█	SIGN	int	
█	APPROVE	char(1)	Nullable
█	CHECKFEE	char(1)	Nullable
█	ISDEFAULT	varchar(1)	Nullable

Unique keys

	Columns	Name / Description
█	IPCTRANCODE	PK_IPCTRANSLIST

Used By

	Name
█	IPCTRANSLIST
⚙	IPC_CHECKTRANAPPROVE

5.1.42. Table: O9\$ERRORCODE

Columns

	Name	Data type	Description / Attributes
	PAGEID	int	
	ITEMID	varchar(50)	
	ORD	int	
	KEY	varchar(50)	Nullable
	VALUE	varchar(150)	Nullable
	ERRDESC	varchar(200)	Nullable
	MANUAL	varchar(5)	Nullable
	ERRDESCFIELD	varchar(200)	Nullable

Unique keys

	Columns	Name / Description
	PAGEID, ITEMID, ORD	PK_DF_MAPPINGERRORCODE

5.1.43. Table: S_SYSTEMCODE

Columns

	Name	Data type	Description / Attributes
☰	cdid 	nvarchar(20)	
☰	cdname	nvarchar(50)	
☰	caption	nvarchar(250)	
☰	cdgrp	nchar(10)	
☰	cdidx	int	
☰	isvisible	bit	

Unique keys

	Columns	Name / Description
	cdid	PK_S_SYSTEMCODE_CDID

Used By

	Name
☰ S_SYSTEMCODE	
⚙ FAST_GETOUTGOING (FAST GET Outgoing Transaction)	
⚙ FAST_GETTRANSDETAIL (FAST Get transaction detail)	

5.1.44. Table: SMP_MASTERPAGE

Columns

	Name	Data type	Description / Attributes
█	masterpageid	int	Identity / Auto increment
█	masterpagename	varchar(50)	
█	masterpagepath	varchar(100)	Nullable
█	masterpageimage	varchar(50)	Nullable
█	masterpagedescription	nvarchar(4000)	Nullable

Linked from

	Table	Join	Title / Name / Description
→	SMP_POSITION	SMP_MASTERPAGE masterpageid = SMP_POSITIONMASTERPAGEID	FK_SMP_Position_SMP_MasterPage1

Unique keys

	Columns	Name / Description
█	masterpageid	PK_SMP_MasterPage

Used By

	Name
█	SMP_MASTERPAGE
⚙️	Pages_GetPageInfo
→	SMP_POSITION

5.1.45. Table: SMP_PAGES

Columns

	Name	Data type	Description / Attributes
PK	pageid	int	Identity / Auto increment
	pagename	nvarchar(200)	
	pagedescription	nvarchar(4000)	Nullable
	portalid	int	
	theme	int	
	pagetitle	nvarchar(1000)	Nullable
	datecreated	datetime	Nullable Default: getdate()
	datemodified	datetime	Nullable
	usermodified	varchar(50)	Nullable
	ishow	bit	Nullable Default: 1
	author	varchar(50)	Nullable
	masterpageid	int	Nullable
	subsystemid	int	Nullable
	trancode	varchar(50)	Nullable
	isdefault	varchar(5)	Nullable Default: 'n'
	isapprove	varchar(5)	Nullable
	linkapprove	varchar(255)	Nullable
	serviceid	varchar(50)	Nullable
	isschedule	varchar(50)	Nullable
	isnotification	varchar(50)	Nullable
	istemplate	varchar(50)	Nullable
	isreceive	varchar(50)	Nullable
	isproductfee	varchar(50)	Nullable
	isreport	varchar(50)	Nullable
	isviewreport	varchar(50)	Nullable
	isbeneficiary	varchar(50)	Nullable
	isreversal	varchar(50)	Nullable

Unique keys

	Columns	Name / Description
PK	pageid	pk_smp_pages

Used By

	Name
PK	SMP_PAGES
FK	IPC_CHECKREVERSAL

Name

PAGES_CHECKPERMIT

Pages_GetPageInfo

POSITION_LOAD

TRIAL

5.1.46. Table: SMP_POSITION

Columns

Name		Data type	Description / Attributes
	POSITIONID	varchar(50)	
	MASTERPAGEID	int	References: SMP_MASTERPAGE

Links to

Table		Join	Title / Name / Description
SMP_MASTERPAGE		SMP_POSITION MASTERPAGEID = SMP_MASTERPAGEmasterpageid	FK_SMP_Position_SMP_MasterPage1

Unique keys

Columns		Name / Description
	POSITIONID, MASTERPAGEID	PK_SMP_Position

Uses

Name	
	SMP_POSITION
SMP_MASTERPAGE	

Used By

Name	
	SMP_POSITION
	POSITION_LOAD

5.1.47. Table: SMP_Settings

Columns

	Name	Data type	Description / Attributes
1	SMTPServer	varchar(50)	Nullable
2	SMTPPort	int	Nullable
3	SMTPUserName	varchar(50)	Nullable
4	SMTPPassword	varchar(50)	Nullable
5	RoleAdminID	int	
6	UserNameDefault	varchar(50)	
7	LogPath	varchar(255)	
8	PortalDefaultID	int	
9	PageDefaultID	int	
10	DefaultLang	varchar(10)	Nullable
11	UserModified	varchar(50)	Nullable
12	DateModified	datetime	Nullable
13	ServiceIDDefault	varchar(50)	Nullable
14	SMTPSSL	bit	Nullable
15	SMTPCCEmail	varchar(50)	Nullable
16	SMTPBCCEmail	varchar(50)	Nullable

Used By

	Name
1	SMP_Settings
2	Settings_LoadPortalSettings

5.1.48. Table: SMP_THEME

Columns

	Name	Data type	Description / Attributes
grid	themeid	int	Identity / Auto increment
grid	themedescription	nvarchar(1000)	Nullable
grid	imagepath	varchar(100)	Nullable
grid	themename	varchar(50)	
grid	usercreated	varchar(50)	Nullable
grid	datecreated	datetime	Nullable Default: getdate()
grid	usermodified	varchar(50)	Nullable
grid	datemodified	datetime	Nullable
grid	filename	nvarchar(50)	Nullable

Unique keys

	Columns	Name / Description
key	themeid	pk_smp_theme

Used By

	Name
grid	SMP_THEME
gear	Pages_GetPageInfo

5.1.49. Table: SMP_USERINROLE

Columns

Name		Data type	Description / Attributes
	USERNAME	varchar(50)	
	ROLEID	int	

Unique keys

Columns		Name / Description
	USERNAME, ROLEID	PK_SMP_UserInRole

Used By

Name	
	SMP_USERINROLE
	PAGES_CHECKPERMIT
	USERS_GETROLE
	WIDGETRIGHT_GETMODULEINPAGEFORROLE

5.1.50. Table: SMP_USERS

Columns

	Name	Data type	Description / Attributes
█	USERNAME	varchar(50)	
█	PASSWORD	varchar(100)	Nullable
█	FIRSTNAME	nvarchar(1000)	
█	MIDDLENAME	nvarchar(1000)	Nullable
█	LASTNAME	nvarchar(1000)	Nullable
█	GENDER	int	Nullable
█	ADDRESS	nvarchar(4000)	Nullable
█	EMAIL	varchar(100)	Nullable
█	BIRTHDAY	datetime	Nullable
█	PHONE	varchar(50)	Nullable
█	LASTLOGINTIME	datetime	Nullable Default: getdate()
█	STATUS	int	Nullable
█	USERCREATED	varchar(50)	Nullable
█	DATECREATED	datetime	Nullable Default: getdate()
█	USERMODIFIED	varchar(50)	Nullable
█	DATEMODIFIED	datetime	Nullable
█	ISLOGIN	varchar(50)	Nullable
█	EXPIRETIME	datetime	Nullable Default: getdate()
█	BRANCHID	varchar(50)	Nullable
█	DEPTID	varchar(50)	Nullable
█	USERLEVEL	int	Nullable
█	USERID	varchar(50)	Nullable
█	COREUSERID	varchar(50)	Nullable
█	CORELGNNAME	varchar(255)	Nullable
█	PRODUCTID	varchar(50)	Nullable
█	ISSHOW	varchar(5)	Nullable Default: 'Y'
█	POLICYID	int	Nullable Default: 0
█	UUID	varchar(50)	Nullable
█	FAILNUMBER	int	Nullable Default: 0

Unique keys

	Columns	Name / Description
█	USERNAME	PK_SMP_Users_1

Used By

Name
SMP_USERS
USERS_GETROLE

TRIAL

5.1.51. Table: SMP_WIDGETPAGE

Columns

	Name	Data type	Description / Attributes
>List	WIDGETPAGEID 	int	Identity / Auto increment
List	PORTALID	int	
List	PAGEID	int	
List	WIDGETID	int	
List	POSITION	varchar(50)	
List	ORDER	int	

Unique keys

	Columns	Name / Description
	WIDGETPAGEID	PK_SMP_WidgetPage

Used By

	Name
	SMP_WIDGETPAGE
	WIDGETRIGHT_GETMODULEINPAGEFORROLE

5.1.52. Table: SMP_WidgetRight

Columns

Name		Data type	Description / Attributes
	WidgetPageID	int	
	RoleID	int	

Unique keys

Columns	Name / Description
WidgetPageID, RoleID	PK_SMP_WidgetRight

Used By

Name
SMP_WidgetRight
WIDGETRIGHT_GETMODULEINPAGEFORROLE

TRIAL

5.1.53. Table: SMP_WIDGETS

Columns

	Name	Data type	Description / Attributes
>List	WIDGETID	int	Identity / Auto increment
List	WIDGETLINK	varchar(200)	
List	DATECREATED	datetime	Nullable Default: getdate()
List	AUTHOR	varchar(50)	Nullable
List	DATEMODIFIED	datetime	Nullable
List	USERMODIFIED	varchar(50)	Nullable
List	ICONPATH	varchar(1000)	Nullable
List	ENABLETHEME	bit	
List	SHOWTITLE	bit	
List	ISPUBLISH	bit	Nullable

Unique keys

	Columns	Name / Description
Key	WIDGETID	PK_SMP_Widgets

Used By

	Name
Table	SMP_WIDGETS
StoredProcedure	WIDGETRIGHT_GETMODULEINPAGEFORROLE
StoredProcedure	WIDGETS_LOAD

5.1.54. Table: SMP_WIDGETSLANG

Columns

Name		Data type	Description / Attributes
WIDGETID	int		
LANGID	varchar(50)		
WIDGETTITLE	nvarchar(4000)	Nullable	

Unique keys

Columns	Name / Description
WIDGETID, LANGID	PK_WidgetsLang

Used By

Name
SMP_WIDGETSLANG
WIDGETRIGHT_GETMODULEINPAGEFORROLE
WIDGETS_LOAD

5.2. Procedures

5.2.1. Procedure: FAST_GET_STATUS

Input/Output

Name	Data type	Description
@StatusID	varchar(50)	

Script

```
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[FAST_GET_STATUS]
    -- Add the parameters for the stored procedure here
    @StatusID varchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    select StatusID,StatusID + ' - ' + StatusName AS StatusName from FAST_STATUS where StatusID=case @StatusID when '' then
StatusID else @StatusID end order by StatusID
END
```

5.2.2. Procedure: FAST_RESEND_TX

Input/Output

	Name	Data type	Description
→@	IPCTRANSID	varchar(1000)	
→@	SENTTOFASTANDCORE	varchar(10)	

Uses

Name
FAST_RESEND_TX
IPCLOGTRANS
IPCSYSVAR
Split

Script

```
-- =====
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[FAST_RESEND_TX]
    -- Add the parameters for the stored procedure here
    @IPCTRANSID VARCHAR(1000),
    @SENTTOFASTANDCORE VARCHAR(10)

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    DECLARE @MAXTRYTOFAST AS INT
    DECLARE @MAXTRYTOCORE AS INT
    SELECT @MAXTRYTOFAST =CAST(VARVALUE AS INT) FROM IPCSYSVAR WHERE VARNAME='TRETRYFAST'
    SELECT @MAXTRYTOCORE =CAST(VARVALUE AS INT) FROM IPCSYSVAR WHERE VARNAME='TRETRY09'
    --INSERT INTO DEVLOG (LOG1,LOG2,LOG3,Time) VALUES (@IPCTRANSID,@ISNOTTOFAST,@ISNOTTOCORE,GETDATE())

    declare @tranid as varchar(50)
    declare @trancode as varchar(100)

    DECLARE db_cursor CURSOR FOR
    SELECT IPCTRANSID,IPCTRANCODE
    FROM IPCLOGTRANS
    WHERE IPCTRANSID IN (SELECT ITEM FROM DBO.Split(@IPCTRANSID,'#'))

    OPEN db_cursor
    FETCH NEXT FROM db_cursor INTO @tranid,@trancode

    WHILE @@FETCH_STATUS = 0
    BEGIN
        if exists (select * from IPCLOGTRANS where IPCTRANSID=@tranid and num67=@MAXTRYTOFAST and num68=1)
            UPDATE IPCLOGTRANS SET NUM67=0 WHERE IPCTRANSID IN (SELECT ITEM FROM DBO.Split(@IPCTRANSID,'#'))
        if exists (select * from IPCLOGTRANS where IPCTRANSID=@tranid and num69=@MAXTRYTOCORE and num70=1)
            UPDATE IPCLOGTRANS SET NUM69=0 WHERE IPCTRANSID IN (SELECT ITEM FROM DBO.Split(@IPCTRANSID,'#'))

        FETCH NEXT FROM db_cursor INTO @tranid,@trancode
    END

    CLOSE db_cursor
    DEALLOCATE db_cursor

END
```

5.2.3. Procedure: FAST_SEARCH_SCHEDULE

Input/Output

	Name	Data type	Description
@	SCHEDULEID	varchar(50)	
@	SCHEDULENAME	varchar(200)	
@	FROM	varchar(20)	
@	TO	varchar(20)	
@	INTERVAL	varchar(10)	
@	DESC	varchar(200)	
@	STATUS	varchar(10)	
@	STARTROWINDEX	int	
@	MAXIMUMROWS	int	

Uses

	Name
⚙️	FAST_SEARCH_SCHEDULE
📄	IPC_SCHEDULEREPEAT (Schedule Repeat)
📄	IPC_SCHEDULES (Schedule Information)

Script

```

CREATE procedure [dbo].[FAST_SEARCH_SCHEDULE]
    @SCHEDULEID VARCHAR(50),
    @SCHEDULENAME VARCHAR(200),
    @FROM VARCHAR(20),
    @TO VARCHAR(20),
    @INTERVAL VARCHAR(10),
    @DESC VARCHAR(200),
    @STATUS VARCHAR(10),
    @STARTROWINDEX INT,
    @MAXIMUMROWS INT
AS
BEGIN
    SET dateformat dMy
    --DECLARE @TRANCODEALL TABLE(TRANCODE VARCHAR(50))
    --INSERT INTO @TRANCODEALL
    --    SELECT TRANCODE FROM SMP_Pages WHERE IsApprove='Y' AND IsShow = 1
    DECLARE @RECORDCOUNT INT = 0
    SELECT @RECORDCOUNT=-
        (SELECT COUNT(1) FROM IPC_SCHEDULES A inner join IPC_SCHEDULEREPEAT B on A.SCHEDULEID=B.SCHEDULEID
        WHERE A.NEXTEXECUTE BETWEEN CAST(@FROM AS datetime) AND DATEADD(DAY,1,CAST(@TO AS datetime))
        AND A.SCHEDULEID LIKE '%' + @SCHEDULEID + '%'
        AND A.SCHEDULENAME LIKE '%' + @SCHEDULEID + '%'
        AND B.DISTANCE = CASE WHEN @INTERVAL='' THEN B.DISTANCE ELSE CAST(@INTERVAL AS INT) END
        AND A.DESCRIPTION LIKE '%' + @SCHEDULEID + '%'
        AND A.STATUS = CASE WHEN @STATUS = 'ALL' THEN A.STATUS ELSE @STATUS END)
    SELECT * FROM (SELECT *,ROW_NUMBER() OVER (ORDER BY scheduleid ) AS ROWRANK, @RECORDCOUNT AS TRECORDCOUNT
    FROM (
        SELECT A.*,B.DISTANCE FROM IPC_SCHEDULES A inner join IPC_SCHEDULEREPEAT B on
        A.SCHEDULEID=B.SCHEDULEID
        WHERE A.NEXTEXECUTE BETWEEN CAST(@FROM AS datetime) AND DATEADD(DAY,1,CAST(@TO AS datetime))
        AND A.SCHEDULEID LIKE '%' + @SCHEDULEID + '%'
        AND A.SCHEDULENAME LIKE '%' + @SCHEDULEID + '%'
        AND B.DISTANCE = CASE WHEN @INTERVAL='' THEN B.DISTANCE ELSE CAST(@INTERVAL AS INT) END
        AND A.DESCRIPTION LIKE '%' + @SCHEDULEID + '%'
        AND A.STATUS = CASE WHEN @STATUS = 'ALL' THEN A.STATUS ELSE @STATUS END
        --AND A.USERCURAPP = CASE WHEN @USERAPP = '' THEN A.USERCURAPP ELSE @USERAPP END
        ) AS TB
        ) AS TRANSACTION_SEARCH
        WHERE RowRank > @STARTROWINDEX AND RowRank <= ( @STARTROWINDEX + @MAXIMUMROWS)
END

```

5.2.4. Procedure: FAST_UPDATE_SCHEDULE

Input/Output

	Name	Data type	Description
→@	SCHEDULEID	varchar(50)	
→@	SCHEDULENAME	varchar(200)	
→@	INTERVAL	varchar(10)	
→@	DESC	varchar(200)	
→@	STATUS	varchar(10)	

Uses

Name
⚙ FAST_UPDATE_SCHEDULE
⌚ IPC_SCHEDULEREPEAT (Schedule Repeat)
⌚ IPC_SCHEDULES (Schedule Information)

Script

```

CREATE procedure [dbo].[FAST_UPDATE_SCHEDULE]
    @SCHEDULEID VARCHAR(50),
    @SCHEDULENAME VARCHAR(200),

    @INTERVAL VARCHAR(10),
    @DESC VARCHAR(200),
    @STATUS VARCHAR(10)

AS
BEGIN

    insert into DEVLOG (log1,log2,log3,Time) values (@SCHEDULEID,@SCHEDULENAME + '/' + @STATUS + '/' + @DESC,@INTERVAL,GETDATE())
    SET dateformat dMy
--DECLARE @TRANCODEALL TABLE(TRANCODE VARCHAR(50))
--INSERT INTO @TRANCODEALL
--    SELECT TRANCODE FROM SMP_Pages WHERE IsApprove='Y' AND IsShow = 1
if(@STATUS='A')
begin
    --update IPC_SCHEDULES set SCHEDULENAME=@SCHEDULENAME,
    DESCRIPTION=@DESC,@STATUS=@STATUS,NEXTEXECUTE=DATEADD(SECOND,cast(@INTERVAL as int),NEXTEXECUTE) where SCHEDULEID=@SCHEDULEID
    update IPC_SCHEDULES set SCHEDULENAME=@SCHEDULENAME, DESCRIPTION=@DESC,STATUS=@STATUS where SCHEDULEID=@SCHEDULEID
    update IPC_SCHEDULEREPEAT set DISTANCE=cast(@INTERVAL as int) where SCHEDULEID=@SCHEDULEID
end
if(@STATUS='D')
begin
    update IPC_SCHEDULES set SCHEDULENAME=@SCHEDULENAME, DESCRIPTION=@DESC,STATUS=@STATUS where SCHEDULEID=@SCHEDULEID
    update IPC_SCHEDULEREPEAT set DISTANCE=cast(@INTERVAL as int) where SCHEDULEID=@SCHEDULEID
end

END

```

5.2.5. Procedure: IPC_CHECKAPPROVED_SCHEDULE

Input/Output

Name	Data type	Description
@SCHEDULEID	varchar(50)	

Uses

Name
IPC_CHECKAPPROVED_SCHEDULE
IPC_SCHEDULES (Schedule Information)
IPC_SCHEDULESHIS (Schedule History)

Script

```
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[IPC_CHECKAPPROVED_SCHEDULE]
    @SCHEDULEID VARCHAR(50)
AS
BEGIN
    if (exists(select * from IPC_SCHEDULES where SCHEDULEID=@SCHEDULEID and ISAPPROVED='Y')
        or exists(select * from IPC_SCHEDULESHIS where SCHEDULEID=@SCHEDULEID and ISAPPROVED='Y'))
begin
    SELECT '0'  ErrorCode , 'OK'  ErrorDesc
end
END
```

5.2.6. Procedure: IPC_CHECKDESTTRANS

Input/Output

	Name	Data type	Description
@	IPCTRANSID	varchar(50)	
@	SOURCEID	varchar(50)	
@	DESTID	varchar(50)	
@	IPCTRANCODE	varchar(100)	
@	IPCTRANCODEDEST	varchar(100)	
@	IPCERRORCODE	varchar(50)	
@	ERRORCODE	varchar(50)	
@	CORPID	varchar(50)	
@	SERID	varchar(50)	
@	DATA	ntext	

Uses

Name
IPC_CHECKDESTTRANS
IPCBLSCHECKREVERSAL
IPCBLSREVERSALFLOW
IPCLOGTRANSREVERSAL
IPCSYSVAR

Script

```
CREATE proc [dbo].[IPC_CHECKDESTTRANS]
@IPCTRANSID VARCHAR(50),
@SOURCEID VARCHAR(50),
@DESTID VARCHAR(50),
@IPCTRANCODE VARCHAR(100),
@IPCTRANCODEDEST VARCHAR(100),
@IPCERRORCODE VARCHAR(50),
@ERRORCODE VARCHAR(50), -- ERRORCODE ALREADY <> 0
@CORPID VARCHAR(50),
@SERID VARCHAR(50),
@DATA NTEXT
AS
BEGIN
    /**
     * N - normal - reversal all transactions
     * P - previous - reversal only previous transactions
     * S - smart - reversal current transaction if system error and previous transactions
    */
    DECLARE @TYPE VARCHAR(2)=''
    DECLARE @SYSWORKDATE NVARCHAR(50)
    SELECT @SYSWORKDATE=VARVALUE FROM IPCSYSVAR WHERE VARNAME='IPCWORKDATE'

    IF EXISTS (SELECT * FROM IPCBLSCHECKREVERSAL WHERE IPCTRANCODE=@IPCTRANCODE AND IPCTRANCODEDEST=@IPCTRANCODEDEST
    AND DESTID=@DESTID AND CORPID=@CORPID AND SERID=@SERID AND ERRORCODE='ALL')
        BEGIN
            SET @TYPE = (SELECT TOP(1) TYPE FROM IPCBLSCHECKREVERSAL WHERE IPCTRANCODE=@IPCTRANCODE AND
IPCTRANCODEDEST=@IPCTRANCODEDEST AND DESTID=@DESTID AND CORPID=@CORPID AND SERID=@SERID AND ERRORCODE='ALL')
            END
            ELSE IF EXISTS (SELECT * FROM IPCBLSCHECKREVERSAL WHERE IPCTRANCODE=@IPCTRANCODE AND
IPCTRANCODEDEST=@IPCTRANCODEDEST AND DESTID=@DESTID AND CORPID=@CORPID AND SERID=@SERID AND ERRORCODE = @ERRORCODE)
                BEGIN
                    SET @TYPE = (SELECT TOP(1) TYPE FROM IPCBLSCHECKREVERSAL WHERE IPCTRANCODE=@IPCTRANCODE AND
IPCTRANCODEDEST=@IPCTRANCODEDEST AND DESTID=@DESTID AND CORPID=@CORPID AND SERID=@SERID AND ERRORCODE = @ERRORCODE)
                    END
                    IF @TYPE <> ''
                    BEGIN
                        DECLARE @CORD INT
                        SET @CORD= (SELECT ORD FROM IPCBLSREVERSALFLOW WHERE IPCTRANCODE = @IPCTRANCODE AND
IPCTRANCODEDEST=@IPCTRANCODEDEST AND DESTID=@DESTID AND CORPID=@CORPID AND SERID=@SERID)
                        DECLARE RVS_CURSOR CURSOR FOR
                            SELECT RVSTRANCODE,RVSDESTRANCODE FROM IPCBLSREVERSALFLOW WHERE IPCTRANCODE = @IPCTRANCODE
                            AND CORPID=@CORPID AND SERID=@SERID AND
                            ORD <= CASE
                                WHEN @TYPE = 'N' THEN @CORD
                                WHEN @TYPE='P' THEN @CORD-1
                                WHEN @TYPE='S' AND @ERRORCODE IN ('9999','ERRORCODE','10499') THEN @CORD --SYSTEM
                                WHEN @TYPE='S' AND @ERRORCODE NOT IN ('9999','ERRORCODE','10499') THEN @CORD -1 -
                                ELSE -1 END
                                DECLARE @RVSTRANCODE VARCHAR(50)
                                DECLARE @RVSDESTRANCODE VARCHAR(50)
                                OPEN RVS_CURSOR;
                                FETCH NEXT FROM RVS_CURSOR INTO @RVSTRANCODE,@RVSDESTRANCODE;
                                WHILE @@FETCH_STATUS = 0
                                BEGIN
                                    IF NOT EXISTS(SELECT * FROM IPCLOGTRANSREVERSAL WHERE IPCTRANSID=@IPCTRANSID AND
RVSTRANCODE=@RVSTRANCODE)
                                        BEGIN
                                            INSERT INTO IPCLOGTRANSREVERSAL
                                            VALUES (@IPCTRANSID,@IPCTRANCODE,@RVSTRANCODE,@RVSDESTRANCODE,@SYSWORKDATE,GETDATE(),1,0,NULL,NULL,@DATA)
                                            END
                                            FETCH NEXT FROM RVS_CURSOR INTO @RVSTRANCODE,@RVSDESTRANCODE;
                                            END
                                            CLOSE RVS_CURSOR;
                                            DEALLOCATE RVS_CURSOR;
                                        END
                                    END
```

5.2.7. Procedure: IPC_CHECKREVERSAL

Input/Output

	Name	Data type	Description
→@	IPCTRANSID	nvarchar(50)	
→@	IPCTRANCODE	nvarchar(50)	
→@	DESTERRORCODE	nvarchar(60)	

Uses

	Name
⚙️	IPC_CHECKREVERSAL
📄	IPCLOGTRANSREVERSAL
📄	IPCSYSVAR
📄	SMP_PAGES

Script

```
-- =====
-- Author:          Admin
-- Create date:    14/02/2009
-- Description:    Reversal Transaction
-- =====
CREATE procedure [dbo].[IPC_CHECKREVERSAL]
    @IPCTRANSID NVARCHAR(50),
    @IPCTRANCODE NVARCHAR(50),
    @DESTERRORCODE NVARCHAR(60)
AS
BEGIN
    insert into DEVLOG(Time,Log1,Log2,Log3) values (GETDATE(),@IPCTRANSID,@IPCTRANCODE,@DESTERRORCODE)
    IF (@DESTERRORCODE IN ('9999','DESTERRORCODE'))
        BEGIN
            IF EXISTS (SELECT * FROM SMP_PAGES WHERE TRANCODE=@IPCTRANCODE AND ISREVERSAL='Y')
                BEGIN
                    IF NOT EXISTS (SELECT * FROM IPCLOGTRANSREVERSAL WHERE IPCTRANSID=@IPCTRANSID)
                        BEGIN
                            DECLARE @SYSWORKDATE NVARCHAR(50)
                            SELECT @SYSWORKDATE=VARVALUE FROM IPCSYSVAR WHERE VARNAME='IPCWORKDATE'
                            INSERT INTO
                                IPCLOGTRANSREVERSAL(IPCTRANSID,IPCTRANCODE,IPCWORKDATE,DATETIMEINSERT,PRIORITY,COUNT)
                                VALUES (@IPCTRANSID,@IPCTRANCODE,@SYSWORKDATE,GETDATE(),0,0)
                        END
                END
        END
END
END
```

5.2.8. Procedure: IPC_CHECKTRANAPPROVE

Input/Output

	Name	Data type	Description
→@	USERID	varchar(50)	
→@	TRANCODE	varchar(50)	
→@	AMOUNT	varchar(50)	
→@	CCYID	varchar(50)	

Uses

	Name
⚙️	IPC_CHECKTRANAPPROVE
📄	IPCTRANSLIST

Script

```

CREATE procedure [dbo].[IPC_CHECKTRANAPPROVE]
    -- Add the parameters for the stored procedure here
    @USERID VARCHAR(50), @TRANCODE VARCHAR(50),
    @AMOUNT VARCHAR(50) = NULL, @CCYID VARCHAR(50) = NULL
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    DECLARE @USERTYPE VARCHAR(10)
    DECLARE @USERLEV INT
    DECLARE @CONTRACTNO VARCHAR(50)
    DECLARE @USERAPPSTYPE VARCHAR(50)
    DECLARE @NEXTLEVAPP INT
    SET @NEXTLEVAPP = 0

    DECLARE @USERAPPID VARCHAR(50)
    DECLARE @NEXTUSERAPP VARCHAR(50)

    SELECT @USERTYPE = USERTYPE, @USERLEV = USERLEVEL, @CONTRACTNO = CONTRACTNO
    FROM EBA_USERS WHERE USERID = @USERID
    IF NOT EXISTS(SELECT * FROM IPCTRANSLIST WHERE IPCTRANCODE = @TRANCODE AND APPROVE = 'Y')
        BEGIN
            SELECT '0' ERRORCODE, 'OK' ERRORDESC, @USERLEV USERLEVEL, @NEXTLEVAPP NEXTLEVAPP, '3'
        APPRSTS
        END
        RETURN
    END
    DECLARE @FLAG INT
    SET @FLAG = -1
    --0:CUSTSITE PASS, 1: WAITTING CUSTSITE, 2: WAITTING BANKSITE, 3: PASS ALL

    -- BEGIN CHECK CUSTSITE
    /* Truong hop nay danh rieng AYA, khong can approve
       IF @USERTYPE = 'RP' -- KHACH HANG CA NHAN DONG SO HUU
       BEGIN
           SELECT '9007' ERRORCODE, 'WAITTING PART-OWNER APPROVE' ERRORDESC, @USERLEV USERLEVEL,
    @NEXTLEVAPP NEXTLEVAPP, '1' APPRSTS
           RETURN
       END
    --
    */
    --Truong hop exesssweeping
    IF @TRANCODE = 'IB000216'
    BEGIN
        SET @FLAG = 3
    END
    --O9CONNECT
    ELSE IF EXISTS(SELECT * FROM EBA_Contract A,EBA_Users B,EBA_Product C WHERE A.ContractNo = B.ContractNo AND
    A.ProductID=C.ProductID AND C.CustType='B' AND B.UserID=@USERID)
    BEGIN
        SET @FLAG = 0
    END
    ELSE
    BEGIN
        IF @USERTYPE = 'RP' -- KHACH HANG CA NHAN DONG SO HUU
        BEGIN
            SET @FLAG = 0
        END
    END
    --Truong hop nay danh rieng AYA, khong can approve
    ELSE
        IF @USERTYPE = 'PCO' -- KHACH HANG CA NHAN

```

```

        BEGIN
            SET @FLAG = 0
        END
    ELSE -- KHACH HANG DOANH NGHIEP, BAT BUOC PHAI CO LEV2 VA LEV1 DUYET
        BEGIN
            SET @FLAG = 1
            SELECT TOP 1 @NEXTLEVAPP = USERLEVEL FROM IBS_USERAPPTRANDETAIL
                WHERE APPTRANID IN (SELECT APPTRANID FROM
                    IBS_USERAPPTRAN
                WHERE TRANCODE = @TRANCODE AND
                    CONTRACTNO = @CONTRACTNO
                AND CCYID = @CCYID AND
                (CAST (@AMOUNT AS NUMERIC(18,3)) BETWEEN FROMLIMIT AND TOLIMIT))
                    ORDER BY ORDAPP
                IF @NEXTLEVAPP = 0
                    IF EXISTS (SELECT * FROM EBA_USERS WHERE CONTRACTNO =
                        @CONTRACTNO AND USERLEVEL = 2 and status<>'D')
                        SET @NEXTLEVAPP = 2
                    ELSE
                        SET @NEXTLEVAPP = 1
                END
        --Truong hop nay danh rieng AYA, khong can approve
        --END CHECK CUSTSITE
        --BEGIN CHECK BANKSITE
        IF @FLAG = 0
            BEGIN
                DECLARE @PRODUCTID VARCHAR(50)
                SELECT @PRODUCTID = PRODUCTID FROM EBA_CONTRACT WHERE CONTRACTNO = @CONTRACTNO

                IF EXISTS (SELECT APPTRANID FROM IBS_TELLERAPPTRAN
                    WHERE TRANCODE = @TRANCODE AND PRODUCTID = @PRODUCTID
                    AND CCYID = @CCYID AND (CAST(@AMOUNT AS NUMERIC(18,3)) BETWEEN
                    FROMLIMIT AND TOLIMIT))
                    BEGIN
                        SET @FLAG = 2
                        SELECT TOP 1 @NEXTLEVAPP = USERLEVEL FROM IBS_TELLERAPPTRANDETAIL
                            WHERE APPTRANID IN (SELECT APPTRANID FROM IBS_TELLERAPPTRAN
                                WHERE TRANCODE = @TRANCODE AND PRODUCTID =
                                    @PRODUCTID
                                AND CCYID = @CCYID AND (CAST(@AMOUNT
                                    AS NUMERIC(18,3)) BETWEEN FROMLIMIT AND TOLIMIT))
                                ORDER BY ORDAPP
                    END
                ELSE
                    SET @FLAG = 3
            END
        --END CHECK BANKSITE
        --RETURN RESULT
        IF @FLAG = 1
        BEGIN
            SELECT '9006' ERRORCODE, 'WAITTING USER APPROVE' ERRORDESC, @USERLEV USERLEVEL, @NEXTLEVAPP NEXTLEVAPP, @FLAG
        APPRSTS
            RETURN
        END
        IF @FLAG = 2
        BEGIN
            SELECT '9005' ERRORCODE, 'WAITTING BANK APPROVE' ERRORDESC, @USERLEV USERLEVEL, @NEXTLEVAPP NEXTLEVAPP, @FLAG
        APPRSTS
            RETURN
        END
        IF @FLAG = 3
        BEGIN
            SELECT '0' ERRORCODE , 'OK' ERRORDESC, @USERLEV USERLEVEL, @NEXTLEVAPP NEXTLEVAPP, @FLAG APPRSTS
            RETURN
        END
    END

```

5.2.9. Procedure: IPC_GETDESTTRANREF

Input/Output

Name		Data type	Description
@	SOURCEID	varchar(20)	
@	DESTID	varchar(20)	
@	IPCTRANCODE	varchar(20)	
@	SOURCETRANREF	varchar(50)	

Uses

Name
IPC_GETDESTTRANREF
IPCLOGTRANS

Script

```
-- =====
-- Author:          Admin
-- Create date: 29/01/2010
-- Description:    Get DestTranref
-- =====
CREATE procedure [dbo].[IPC_GETDESTTRANREF]
    @SOURCEID           VARCHAR(20),
    @DESTID             VARCHAR(20),
    @IPCTRANCODE        VARCHAR(20),
    @SOURCETRANREF      VARCHAR(50)

AS
BEGIN
    SELECT * FROM IPCLOGTRANS A (NOLOCK)
    WHERE A.SOURCEID = @SOURCEID AND A.DESTID = @DESTID
        AND A.IPCTRANCODE = @IPCTRANCODE
        AND A.SOURCETRANREF = @SOURCETRANREF
        AND A.STATUS = 1
END
```

5.2.10. Procedure: IPC_GETLISTREVERSAL

Uses

Name
IPC_GETLISTREVERSAL
IPCLOGTRANSREVERSAL
IPCSYSVAR

Script

```
-- =====
-- Author:          Admin
-- Create date: 14/02/2009
-- Description:    Reversal Transaction
-- =====
CREATE procedure [dbo].[IPC_GETLISTREVERSAL]
AS
BEGIN
    DECLARE @SYSWORKDATE NVARCHAR(50)
    SELECT @SYSWORKDATE=VARVALUE FROM IPCSYSVAR WHERE VARNAME='IPCWORKDATE'

    SELECT * FROM IPCLOGTRANSREVERSAL where IPCWORKDATE = @SYSWORKDATE AND COUNT<50
END
```

TRIAL

5.2.11. Procedure: IPC_GETNEXTEXECSCHEDULE

Input/Output

Name	Data type	Description
@SCHEDULEID	varchar(50)	

Uses

Name
IPC_GETNEXTEXECSCHEDULE
IPC_SCHEDULEDAY
IPC_SCHEDULEREPEAT (Schedule Repeat)
IPC_SCHEDULES (Schedule Information)

Script

```

CREATE procedure [dbo].[IPC_GETNEXTEXECSCHEDULE]
    -- Add the parameters for the stored procedure here
    @SCHEDULEID VARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @TYPE VARCHAR(50)
    DECLARE @NEXT DATETIME
    DECLARE @CURDAY INT
    DECLARE @NEXTDAY INT
    DECLARE @NEXE DATETIME
    DECLARE @DISTANCE INT=0
    SELECT @NEXE = SCHEDULETIME FROM IPC_SCHEDULES WHERE SCHEDULEID = @SCHEDULEID
    select @TYPE=scheduleType from IPC_SCHEDULES where scheduleID=@SCHEDULEID
    UPDATE IPC_SCHEDULES SET NEXTEXECUTE = SCHEDULETIME WHERE SCHEDULEID = @SCHEDULEID
    IF(@TYPE='REPEATEDLY')
        BEGIN
            SELECT @DISTANCE=DISTANCE FROM IPC_SCHEDULEREPEAT WHERE SCHEDULEID=@SCHEDULEID
            UPDATE IPC_SCHEDULES SET NEXTEXECUTE = DATEADD(SECOND,@DISTANCE ,GETDATE()) WHERE SCHEDULEID
            = @SCHEDULEID
            RETURN
        END
    IF(@TYPE = 'ONETIME' OR @TYPE = 'DAILY')
        BEGIN
            UPDATE IPC_SCHEDULES SET NEXTEXECUTE = SCHEDULETIME WHERE SCHEDULEID = @SCHEDULEID
            RETURN
        END
    IF(@TYPE = 'WEEKLY')
        BEGIN
            SELECT @CURDAY = DATEPART(DW, @NEXE)
            IF(EXISTS(SELECT DAYNO FROM IPC_SCHEDULEDAY WHERE SCHEDULEID = @SCHEDULEID AND DAYNO =
            @CURDAY)
                AND @NEXE > GETDATE())
                BEGIN
                    RETURN
                END
            ELSE
                BEGIN
                    SET @NEXTDAY = -1
                    SELECT TOP 1 @NEXTDAY = DAYNO FROM IPC_SCHEDULEDAY
                    WHERE SCHEDULEID = @SCHEDULEID AND DAYNO >
                    @CURDAY ORDER BY DAYNO
                    IF(@NEXTDAY = -1)
                        BEGIN
                            SELECT TOP 1 @NEXTDAY = DAYNO FROM
                            WHERE SCHEDULEID = @SCHEDULEID ORDER
                            SELECT @NEXTDAY = @NEXTDAY + 7 - @CURDAY
                        END
                    ELSE
                        BEGIN
                            SELECT @NEXTDAY = @NEXTDAY - @CURDAY
                        END
                    UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                    DATEADD(DAY,@NEXTDAY,NEXTEXECUTE) WHERE SCHEDULEID = @SCHEDULEID
                END
            END
        IF(@TYPE = 'MONTHLY')
            BEGIN
                SELECT @CURDAY = DATEPART(DAY,@NEXE)
            END
        
```

```

        IF(EXISTS (SELECT DAYNO FROM IPC_SCHEDULEDAY WHERE SCHEDULEID = @SCHEDULEID AND DAYNO =
@CURDAY)
                AND @NEXE > GETDATE())
        BEGIN
                RETURN
        END
        ELSE
        BEGIN
                SET @NEXTDAY = -1
                SELECT TOP 1 @NEXTDAY = DAYNO FROM IPC_SCHEDULEDAY
                WHERE SCHEDULEID = @SCHEDULEID AND DAYNO >
@CURDAY ORDER BY DAYNO
                IF(@NEXTDAY = -1)
                BEGIN
                        SELECT TOP 1 @NEXTDAY = DAYNO FROM
                                WHERE SCHEDULEID = @SCHEDULEID ORDER
                UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                SELECT @NEXTDAY = @NEXTDAY - @CURDAY
                UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                DATEADD(DAY, @NEXTDAY, NEXTEXECUTE) WHERE SCHEDULEID = @SCHEDULEID
                END
                ELSE
                BEGIN
                        SET @NEXTDAY = -1
                        SELECT TOP 1 @NEXTDAY = DAYNO FROM
                                WHERE SCHEDULEID = @SCHEDULEID ORDER
                UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                SELECT @NEXTDAY = @NEXTDAY - @CURDAY
                UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                DATEADD(DAY, @NEXTDAY, NEXTEXECUTE) WHERE SCHEDULEID = @SCHEDULEID
                END
                END
        END

```

TRIAL

5.2.12. Procedure: IPC_GETNEXTTIMEEXEC_bk20171121

Uses

Name
IPC_GETNEXTTIMEEXEC_bk20171121
IPC_SCHEDULES (Schedule Information)

Script

```
create procedure [dbo].[IPC_GETNEXTTIMEEXEC_bk20171121]
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE IPC_SCHEDULES
    SET NEXTEXECUTE=DATEADD(SECOND,3,GETDATE())
    WHERE NEXTEXECUTE <= GETDATE() AND STATUS = 'A'

    DECLARE @MINN DATETIME = (SELECT MIN(NEXTEXECUTE) FROM IPC_SCHEDULES WHERE STATUS = 'A')
    SELECT CASE WHEN @MINN > GETDATE() THEN @MINN - GETDATE() ELSE DATEADD(SECOND,3,GETDATE()) -GETDATE() END NEXTEXECUTE
END
```

TRIAL

5.2.13. Procedure: IPC_GETTRANSID_TEMP

Uses

Name
IPC_GETTRANSID_TEMP
IPCSYSVAR

Script

```
-- =====
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE procedure [dbo].[IPC_GETTRANSID_TEMP]
    -- Add the parameters for the stored procedure here
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    BEGIN TRAN
    BEGIN TRY
        UPDATE IPCSYSVAR SET VARVALUE = CAST(CAST(VARVALUE AS NUMERIC) + 1 AS VARCHAR)
        WHERE VARNAME = 'IPCTRANSID'
        SELECT VARVALUE FROM IPCSYSVAR WHERE VARNAME = 'IPCTRANSID'
        COMMIT TRAN
    END TRY
    BEGIN CATCH
        ROLLBACK TRAN
    END CATCH
END
```

5.2.14. Procedure: IPC_GETURL_BYSOURCETRANCODE

Input/Output

Name	Data type	Description
@IPCTRANCODE	varchar(100)	

Uses

Name
IPC_GETURL_BYSOURCETRANCODE
IPCCOMPONENT (IPC component)
IPCCONNECTIONWS (IPC Connection WebService)

Script

```
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[IPC_GETURL_BYSOURCETRANCODE]
    -- Add the parameters for the stored procedure here
    @IPCTRANCODE varchar(100)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    select * from IPCCONNECTIONWS a
inner join IPCCOMPONENT b on a.IPCTRANCODE=b.PARMLIST
where b.IPCTRANCODE=@IPCTRANCODE
END
```

5.2.15. Procedure: IPC_GetWorkingDate

Uses

Name
 IPC_GetWorkingDate
 IPCSYSVAR

Script

```
-- =====
-- Author:          Admin
-- Create date: <Create Date,,>
-- Description:    Lay workingdate
-- =====
CREATE procedure [dbo].[IPC_GetWorkingDate]
    -- Add the parameters for the stored procedure here

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT * FROM IPCSYSVAR WHERE VARNAME='IPCWORKDATE'
END
```

TRIAL

5.2.16. Procedure: IPC_RESETID

Uses

Name
IPC_RESETID
IPCINVENTORY

Script

```
-- =====
-- Author:      Admin
-- Create date: 14/02/2009
-- Description: RESET ID
-- =====
CREATE procedure [dbo].[IPC_RESETID]
AS
BEGIN
    BEGIN TRAN RESETID
    UPDATE IPCINVENTORY SET SEQVALUE = 1 WHERE SEQNAME = 'SMBTXNUM'

    IF @@ERROR = 0
    BEGIN
        COMMIT TRAN RESETID
        SELECT '1' RESULT
    END
    ELSE
    BEGIN
        ROLLBACK TRAN RESETID
        SELECT '0' RESULT
    END
END
```

TRIAL

5.2.17. Procedure: IPC_SCHEDULELOADACTION

Uses

Name
 IPC_SCHEDULELOADACTION
 IPC_SCHEDULEACTION (Schedule Action)

Script

```
CREATE procedure [dbo].[IPC_SCHEDULELOADACTION]
    -- Add the parameters for the stored procedure here
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT * FROM IPC_SCHEDULEACTION
END
```

TRIAL

5.2.18. Procedure: IPC_SCHEDULESGETCURRENT

Uses

Name
IPC_SCHEDULESGETCURRENT
IPC_SCHEDULEDAY
IPC_SCHEDULEDAYHIS
IPC_SCHEDULEDETAIL (Schedule Detail)
IPC_SCHEDULEDETAILHIS (Schedule detail history)
IPC_SCHEDULEREPEAT (Schedule Repeat)
IPC_SCHEDULEREPEATHIS
IPC_SCHEDULES (Schedule Information)
IPC_SCHEDULESHIS (Schedule History)

Script

```

CREATE procedure [dbo].[IPC_SCHEDULESGETCURRENT]
AS
BEGIN

    SET NOCOUNT ON;
    SET NOCOUNT ON;
    DECLARE @ID VARCHAR(50)
    DECLARE @TYPE VARCHAR(50)
    DECLARE @NEXT DATETIME
    DECLARE @CURDAY INT
    DECLARE @NEXTDAY INT
    DECLARE @NOW DATETIME
    SET @NOW = GETDATE()

    DECLARE @DISTANCE NUMERIC(18, 0)
    DECLARE @NRETRY INT
    DECLARE @TRETRY INT
    DECLARE @ENDDATE DATETIME

    SELECT * FROM IPC_SCHEDULES WHERE NEXTEXECUTE <= @NOW AND STATUS = 'A' ORDER BY ORD

    SELECT * FROM IPC_SCHEDULEDETAIL ISL WHERE ISL.SCHEDULEID IN (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE
    NEXTEXECUTE <= @NOW AND STATUS = 'A')

    DECLARE SCHEDULE_CURSOR CURSOR FOR
        SELECT SCHEDULEID, SCHEDULETYPE, NEXTEXECUTE FROM IPC_SCHEDULES WHERE (DATEDIFF(DAY,NEXTEXECUTE,@NOW)=0
    AND NEXTEXECUTE <= @NOW) AND STATUS = 'A'

    OPEN SCHEDULE_CURSOR
    FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE, @NEXT
    WHILE (@@FETCH_STATUS=0)
        BEGIN
            IF(@TYPE = 'DAILY')
                BEGIN
                    UPDATE IPC_SCHEDULES SET NEXTEXECUTE = DATEADD(DAY, 0, DATEDIFF(DAY,
0, DATEADD(DAY,1,NEXTEXECUTE))) + DATEADD(DAY, 0 - DATEDIFF(DAY, 0, SCHEDULETIME), SCHEDULETIME) WHERE SCHEDULEID = @ID
                    FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE, @NEXT
                    CONTINUE
                END

            IF(@TYPE = 'REPEATEDLY')
                BEGIN
                    SELECT @DISTANCE=DISTANCE, @NRETRY=NRETRY, @TRETRY=TRETRY FROM
                    IPC_SCHEDULEREPEAT WHERE SCHEDULEID=@ID
                    SELECT @ENDDATE=ENDDATE FROM IPC_SCHEDULES WHERE SCHEDULEID=@ID
                    IF((@NRETRY < @TRETRY OR @TRETRY < 0) AND @NOW < @ENDDATE)
                        BEGIN
                            UPDATE IPC_SCHEDULEREPEAT SET NRETRY= CASE WHEN
                            @TRETRY < 0 THEN -1 ELSE @NRETRY + 1 END WHERE SCHEDULEID=@ID
                            UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                            DATEADD(SECOND,@DISTANCE,@NOW) WHERE SCHEDULEID = @ID
                            @NEXT
                        END
                    ELSE
                        BEGIN
                            UPDATE IPC_SCHEDULES SET STATUS = 'Y' WHERE
                        END
                END
            END
        END
    END

```

```

SCHEDULEID = @ID
@NEXT
        FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE,
        CONTINUE
    END
    IF (@TYPE = 'WEEKLY')
        BEGIN
            SELECT @CURDAY = DATEPART(DW, @NOW)
            SET @NEXTDAY = -1
            SELECT TOP 1 @NEXTDAY = DAYNO FROM IPC_SCHEDULEDAY
                WHERE SCHEDULEID = @ID AND DAYNO > @CURDAY ORDER
BY DAYNO
                BY DAYNO
                IF (@NEXTDAY = -1)
                    BEGIN
                        SELECT TOP 1 @NEXTDAY = DAYNO FROM
                            WHERE SCHEDULEID = @ID ORDER BY DAYNO
                        SELECT @NEXTDAY = @NEXTDAY + 7 - @CURDAY
                    END
                ELSE
                    BEGIN
                        SELECT @NEXTDAY = @NEXTDAY - @CURDAY
                    END
                UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
DATEADD(DAY, @NEXTDAY, NEXTEXECUTE) WHERE SCHEDULEID = @ID
                FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE, @NEXT
                CONTINUE
            END
            IF (@TYPE = 'MONTHLY')
                BEGIN
                    SELECT @CURDAY = DATEPART(DAY, @NOW)
                    SET @NEXTDAY = -1
                    SELECT TOP 1 @NEXTDAY = DAYNO FROM IPC_SCHEDULEDAY
                        WHERE SCHEDULEID = @ID AND DAYNO > @CURDAY ORDER
BY DAYNO
                    IF (@NEXTDAY = -1)
                        BEGIN
                            SELECT TOP 1 @NEXTDAY = DAYNO FROM
                                WHERE SCHEDULEID = @ID ORDER BY DAYNO
                            UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                            UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                        END
                    ELSE
                        BEGIN
                            SELECT @NEXTDAY = @NEXTDAY - @CURDAY
                            UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                        END
                    UPDATE IPC_SCHEDULES SET STATUS = 'Y' WHERE SCHEDULEID = @ID
                    FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE, @NEXT
                    CONTINUE
                END
                IF (@TYPE = 'ONETIME')
                    BEGIN
                        UPDATE IPC_SCHEDULES SET STATUS = 'Y' WHERE SCHEDULEID = @ID
                        FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE, @NEXT
                    END
                FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE, @NEXT
            END
        CLOSE SCHEDULE_CURSOR
        DEALLOCATE SCHEDULE_CURSOR
        UPDATE IPC_SCHEDULES SET STATUS = 'Y' WHERE NEXTEXECUTE >= DATEADD(DAY, 1, ENDDATE)

        INSERT INTO IPC_SCHEDULESHIS
        SELECT *, @NOW FROM IPC_SCHEDULES WHERE STATUS = 'Y'
        INSERT INTO IPC_SCHEDULEDAYHIS
        SELECT * FROM IPC_SCHEDULEDAY WHERE SCHEDULEID IN
            (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
        INSERT INTO IPC_SCHEDULEDETAILHIS
        SELECT * FROM IPC_SCHEDULEDETAIL WHERE SCHEDULEID IN
            (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
        INSERT INTO IPC_SCHEDULEREPEATHIS
        SELECT * FROM IPC_SCHEDULEREPEAT WHERE SCHEDULEID IN
            (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS='Y')

        DELETE IPC_SCHEDULEDAY WHERE SCHEDULEID IN
            (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
        DELETE IPC_SCHEDULEDETAIL WHERE SCHEDULEID IN
            (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
        DELETE IPC_SCHEDULEREPEAT WHERE SCHEDULEID IN
            (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
        DELETE IPC_SCHEDULES WHERE STATUS = 'Y'

```

TRIAL

5.2.19. Procedure: IPC_SCHEDULESLOGIN

Uses

Name
IPC_SCHEDULESLOGIN
IPC_SCHEDULEDAY
IPC_SCHEDULEDAYHIS
IPC_SCHEDULEDETAIL (Schedule Detail)
IPC_SCHEDULEDETAILHIS (Schedule detail history)
IPC_SCHEDULEREPEAT (Schedule Repeat)
IPC_SCHEDULEREPEATHIS
IPC_SCHEDULES (Schedule Information)
IPC_SCHEDULESHIS (Schedule History)

Script

```
CREATE PROCEDURE [dbo].[IPC_SCHEDULESLOGIN]
AS
BEGIN

    SET NOCOUNT ON;
    SET NOCOUNT ON;
    DECLARE @ID VARCHAR(50)
    DECLARE @TYPE VARCHAR(50)
    DECLARE @NEXT DATETIME
    DECLARE @CURDAY INT
    DECLARE @NEXTDAY INT
    DECLARE @NOW DATETIME
    SET @NOW = GETDATE()

    DECLARE @DISTANCE NUMERIC(18, 0)
    DECLARE @NRETRY INT
    DECLARE @TRETRY INT
    DECLARE @ENDDATE DATETIME

    SELECT * FROM IPC_SCHEDULES WHERE STATUS = 'A' AND SCHEDULEID = 'SCHLOGINO9' ORDER BY ORD
    SELECT * FROM IPC_SCHEDULEDETAIL WHERE SCHEDULEID IN (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'A' AND SCHEDULEID = 'SCHLOGINO9')

    DECLARE SCHEDULE_CURSOR CURSOR FOR
        SELECT SCHEDULEID, SCHEDULETYPE, NEXTEXECUTE FROM IPC_SCHEDULES
        WHERE STATUS = 'A' AND SCHEDULEID = 'SCHLOGINO9' ORDER BY ORD

    OPEN SCHEDULE_CURSOR
    FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE, @NEXT
    WHILE (@@FETCH_STATUS=0)
    BEGIN
        IF(@TYPE = 'DAILY')
        BEGIN
            UPDATE IPC_SCHEDULES SET NEXTEXECUTE = DATEADD(DAY, 0, DATEDIFF(DAY, 0, DATEADD(DAY, 1, NEXTEXECUTE)) + DATEADD(DAY, 0 - DATEDIFF(DAY, 0, SCHEUDLETIME), SCHEUDLETIME)) WHERE SCHEDULEID = @ID
            FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE, @NEXT
            CONTINUE
        END

        IF(@TYPE = 'REPEATEDLY')
        BEGIN
            SELECT @DISTANCE=DISTANCE, @NRETRY=NRETRY, @TRETRY=TRETRY FROM
            IPC_SCHEDULEREPEAT WHERE SCHEDULEID=@ID
            SELECT @ENDDATE=ENDDATE FROM IPC_SCHEDULES WHERE SCHEDULEID=@ID
            IF((@NRETRY < @TRETRY OR @TRETRY < 0) AND @NOW < @ENDDATE)
            BEGIN
                UPDATE IPC_SCHEDULEREPEAT SET NRETRY= CASE WHEN
                @TRETRY < 0 THEN -1 ELSE @NRETRY + 1 END WHERE SCHEDULEID=@ID
                UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                DATEADD(SECOND, @DISTANCE, @NOW) WHERE SCHEDULEID = @ID
                @NEXT
            END
        ELSE
    END

```

```

BEGIN
    UPDATE IPC_SCHEDULES SET STATUS = 'Y' WHERE
        FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE,
        CONTINUE
    END
END
IF (@TYPE = 'WEEKLY')
BEGIN
    SELECT @CURDAY = DATEPART(DW, @NOW)
    SET @NEXTDAY = -1
    SELECT TOP 1 @NEXTDAY = DAYNO FROM IPC_SCHEDULEDAY
        WHERE SCHEDULEID = @ID AND DAYNO > @CURDAY ORDER
        BY DAYNO
    IF (@NEXTDAY = -1)
        BEGIN
            SELECT TOP 1 @NEXTDAY = DAYNO FROM
                WHERE SCHEDULEID = @ID ORDER BY DAYNO
            SELECT @NEXTDAY = @NEXTDAY + 7 - @CURDAY
        END
    ELSE
        BEGIN
            SELECT @NEXTDAY = @NEXTDAY - @CURDAY
        END
    UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
        DATEADD(DAY, @NEXTDAY, NEXTEXECUTE) WHERE SCHEDULEID = @ID
        FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE, @NEXT
        CONTINUE
    IF (@TYPE = 'MONTHLY')
        BEGIN
            SELECT @CURDAY = DATEPART(DAY, @NOW)
            SET @NEXTDAY = -1
            SELECT TOP 1 @NEXTDAY = DAYNO FROM IPC_SCHEDULEDAY
                WHERE SCHEDULEID = @ID AND DAYNO > @CURDAY ORDER
                BY DAYNO
            IF (@NEXTDAY = -1)
                BEGIN
                    SELECT TOP 1 @NEXTDAY = DAYNO FROM
                        WHERE SCHEDULEID = @ID ORDER BY DAYNO
                    UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                        UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                END
            ELSE
                BEGIN
                    SELECT @NEXTDAY = @NEXTDAY - @CURDAY
                    UPDATE IPC_SCHEDULES SET NEXTEXECUTE =
                END
            UPDATE IPC_SCHEDULES SET STATUS = 'Y' WHERE SCHEDULEID = @ID
            FETCH NEXT FROM SCHEDULE_CURSOR INTO @ID, @TYPE, @NEXT
            CONTINUE
        END
    IF (@TYPE = 'ONETIME')
        BEGIN
            DELETE IPC_SCHEDULEDAY WHERE SCHEDULEID IN
                (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
            DELETE IPC_SCHEDULEDETAIL WHERE SCHEDULEID IN
                (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
            DELETE IPC_SCHEDULEREPEAT WHERE SCHEDULEID IN
                (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
            DELETE IPC_SCHEDULESHIS
            SELECT *, @NOW FROM IPC_SCHEDULES WHERE STATUS = 'Y'
            INSERT INTO IPC_SCHEDULESHIS
            SELECT * FROM IPC_SCHEDULEDAY WHERE SCHEDULEID IN
                (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
            INSERT INTO IPC_SCHEDULEDETAILHIS
            SELECT * FROM IPC_SCHEDULEDETAIL WHERE SCHEDULEID IN
                (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
            INSERT INTO IPC_SCHEDULEREPEATHIS
            SELECT * FROM IPC_SCHEDULEREPEAT WHERE SCHEDULEID IN
                (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS='Y')
        END
    CLOSE SCHEDULE_CURSOR
    DEALLOCATE SCHEDULE_CURSOR
    UPDATE IPC_SCHEDULES SET STATUS = 'Y' WHERE NEXTEXECUTE >= DATEADD(DAY, 1, ENDDATE)
    INSERT INTO IPC_SCHEDULESHIS
    SELECT *, @NOW FROM IPC_SCHEDULES WHERE STATUS = 'Y'
    INSERT INTO IPC_SCHEDULEDAYHIS
    SELECT * FROM IPC_SCHEDULEDAY WHERE SCHEDULEID IN
        (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
    INSERT INTO IPC_SCHEDULEDETAILHIS
    SELECT * FROM IPC_SCHEDULEDETAIL WHERE SCHEDULEID IN
        (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
    INSERT INTO IPC_SCHEDULEREPEATHIS
    SELECT * FROM IPC_SCHEDULEREPEAT WHERE SCHEDULEID IN
        (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS='Y')
    DELETE IPC_SCHEDULEDAY WHERE SCHEDULEID IN
        (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
    DELETE IPC_SCHEDULEDETAIL WHERE SCHEDULEID IN
        (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
    DELETE IPC_SCHEDULEREPEAT WHERE SCHEDULEID IN
        (SELECT SCHEDULEID FROM IPC_SCHEDULES WHERE STATUS = 'Y')
    DELETE IPC_SCHEDULES WHERE STATUS = 'Y'
END

```

TRIAL

5.2.20. Procedure: IPC_TIMEOUTTRANS_SELECT

Uses

Name
 IPC_TIMEOUTTRANS_SELECT
 IPC_TIMEOUTTRANS

Script

```
CREATE proc [dbo].[IPC_TIMEOUTTRANS_SELECT]
AS
BEGIN
    SELECT * FROM IPC_TIMEOUTTRANS
END
```

TRIAL

5.2.21. Procedure: IPC_UPDATE_SESSION

Input/Output

	Name	Data type	Description
→@	USERNAME	varchar(50)	
→@	SESSIONID	varchar(50)	
→@	BRANCHCD	varchar(4000)	
→@	BUSDATE	varchar(20)	

Uses

	Name
⚙️	IPC_UPDATE_SESSION
📄	IPCSYSVAR
📄	S_USRSSN (User session)

Script

```
-- =====
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[IPC_UPDATE_SESSION]
    -- Add the parameters for the stored procedure here
    @USERNAME VARCHAR(50),
    @SESSIONID VARCHAR(50),
    @BRANCHCD VARCHAR(4000),
    @BUSDATE VARCHAR(20)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    IF NOT EXISTS (SELECT * FROM dbo.S_USRSSN WHERE username=@USERNAME)
        INSERT INTO dbo.S_USRSSN(username,password,sessionid,busdate,branchcd)
VALUES (@USERNAME, '', @SESSIONID, @BRANCHCD, @BUSDATE);
    ELSE
        UPDATE dbo.S_USRSSN SET sessionid = @SESSIONID , branchcd = @BRANCHCD , busdate = CAST(@BUSDATE AS DATE) WHERE
username = @USERNAME;

        UPDATE IPCSYSVAR SET VARVALUE = @BUSDATE WHERE VARNAME = 'IPCWORKDATE' ;

END
```

5.2.22. Procedure: IPC_UPDATEBATCHTRANS

Input/Output

	Name	Data type	Description
→@	IPCTRANSID	numeric(18, 0)	
→@	TRANDESC	nvarchar(100)	
→@	SEDFEE	numeric(18, 2)	
→@	REVFEE	numeric(18, 2)	

Uses

	Name
⚙️	IPC_UPDATEBATCHTRANS
📄	IPCLOGTRANS

Script

```
CREATE procedure [dbo].[IPC_UPDATEBATCHTRANS]
    @IPCTRANSID numeric
    ,@TRANDESC nvarchar(100)
    ,@SEDFEE numeric(18,2)
    ,@REVFEE numeric(18,2)
AS
BEGIN
    UPDATE [dbo].[IPCLOGTRANS]
    SET
        TRANDESC = CASE WHEN @TRANDESC = '' THEN TRANDESC ELSE @TRANDESC END
        ,[NUM02] = CASE WHEN @SEDFEE = 0 THEN NUM02 ELSE @SEDFEE END
        ,[NUM03] = CASE WHEN @REVFEE = 0 THEN NUM03 ELSE @REVFEE END
    WHERE IPCTRANSID = @IPCTRANSID
END
```

5.2.23. Procedure: IPC_UPDATELISTREVERSAL

Input/Output

	Name	Data type	Description
→@	TRANREF	nvarchar(50)	
→@	TRANCODE	nvarchar(200)	
→@	ERRORCODE	nvarchar(200)	
→@	ERRORDESC	nvarchar(200)	

Uses

	Name
⚙️	IPC_UPDATELISTREVERSAL
📄	IPCLOGTRANS
📄	IPCLOGTRANSREVERSAL
📄	IPCLOGTRANSREVERSALHIS

TRIAL

Script

```
-- =====
-- Author: Admin
-- Create date: 14/02/2009
-- Description: Reversal Transaction
-- =====
CREATE procedure [dbo].[IPC_UPDATERLISTREVERSAL]
    @TRANREF NVARCHAR(50),
    @TRANCODE NVARCHAR(200),
    @ERRORCODE NVARCHAR(200),
    @ERRORDESC NVARCHAR(200)
AS
BEGIN
DECLARE @COUNT INT =0
---27612-27112: already reversal
---26583: invalid transaction
    --IF(@ERRORCODE IN ('0','-27612','-27112','-26583','-26612'))
        IF EXISTS(SELECT * FROM BLs_RVSERRORLIST WHERE IPCTRANCODE=@TRANCODE AND ERRORCODE = @ERRORCODE AND STATUS = 'A' AND TYPE='S')
            BEGIN
                IF EXISTS(SELECT * FROM IPCLOGTRANSREVERSAL WHERE IPCTRANSID=@TRANREF)
                    BEGIN
                        INSERT INTO
IPCLOGTRANSREVERSALHIS(IPCTRANSID,IPCTRANCODE,RVSTRANCODE,RVSDESTID,IPCWORKDATE,DATETIMEINSERT,PRIORITY,COUNT,DATETIMEPROCESS
,ERRORCODE,ERRORDESC,DATA)
SELECT
IPCTRANSID,IPCTRANCODE,RVSTRANCODE,RVSDESTID,IPCWORKDATE,DATETIMEINSERT,PRIORITY,COUNT,GETDATE(),@ERRORCODE,@ERRORDESC,DATA
FROM IPCLOGTRANSREVERSAL
WHERE IPCTRANSID=@TRANREF AND RVSTRANCODE=@TRANCODE
DELETE FROM IPCLOGTRANSREVERSAL WHERE IPCTRANSID=@TRANREF AND
RVSTRANCODE=@TRANCODE
                    END
                ELSE
                    BEGIN
                        INSERT INTO
IPCLOGTRANSREVERSALHIS(IPCTRANSID,IPCTRANCODE,IPCWORKDATE,DATETIMEINSERT,PRIORITY,COUNT,DATETIMEPROCESS,ERRORCODE,ERRORDESC)
VALUES (@TRANREF,'','','GETDATE(),0,0,GETDATE(),@ERRORCODE,@ERRORDESC)
                    END
                IF NOT EXISTS(SELECT * FROM IPCLOGTRANSREVERSAL WHERE IPCTRANSID = @TRANREF)
                    UPDATE IPCLOGTRANS SET DELETED = 1 WHERE IPCTRANSID=@TRANREF
                END
            ELSE
                BEGIN
                    IF EXISTS(SELECT * FROM IPCLOGTRANSREVERSAL WHERE IPCTRANSID=@TRANREF)
                        BEGIN
                            SELECT @COUNT=COUNT FROM IPCLOGTRANSREVERSAL WHERE IPCTRANSID=@TRANREF
                            UPDATE IPCLOGTRANSREVERSAL
                            SET COUNT=@COUNT+1,ERRORCODE=@ERRORCODE,ERRORDESC=@ERRORDESC
                            WHERE IPCTRANSID=@TRANREF AND RVSTRANCODE=@TRANCODE
                        END
                    ELSE
                        BEGIN
                            INSERT INTO
IPCLOGTRANSREVERSALHIS(IPCTRANSID,IPCTRANCODE,IPCWORKDATE,DATETIMEINSERT,PRIORITY,COUNT,DATETIMEPROCESS,ERRORCODE,ERRORDESC)
VALUES (@TRANREF,'','','GETDATE(),0,1,GETDATE(),@ERRORCODE,@ERRORDESC)
                        END
                END
            END
        END
END
```

5.2.24. Procedure: IPC_UPDATELOGDEFINE

Input/Output

	Name	Data type	Description
@	IPCTRANSID	numeric(18, 4)	
@	CHAR01	nvarchar(1000)	
@	CHAR02	nvarchar(1000)	
@	CHAR03	nvarchar(1000)	
@	CHAR04	nvarchar(1000)	
@	CHAR05	nvarchar(1000)	
@	CHAR06	nvarchar(1000)	
@	CHAR07	nvarchar(1000)	
@	CHAR08	nvarchar(1000)	
@	CHAR09	nvarchar(1000)	
@	CHAR10	nvarchar(1000)	
@	CHAR11	nvarchar(1000)	
@	CHAR12	nvarchar(1000)	
@	CHAR13	nvarchar(1000)	
@	CHAR14	nvarchar(1000)	
@	CHAR15	nvarchar(1000)	
@	CHAR16	nvarchar(1000)	
@	CHAR17	nvarchar(1000)	
@	CHAR18	nvarchar(1000)	
@	CHAR19	nvarchar(1000)	
@	CHAR20	nvarchar(1000)	
@	CHAR21	nvarchar(1000)	
@	CHAR22	nvarchar(1000)	
@	CHAR23	nvarchar(1000)	
@	CHAR24	nvarchar(1000)	
@	CHAR25	nvarchar(1000)	
@	CHAR26	nvarchar(1000)	
@	CHAR27	nvarchar(1000)	
@	CHAR28	nvarchar(1000)	
@	CHAR29	nvarchar(1000)	
@	CHAR30	nvarchar(1000)	
@	CHAR31	nvarchar(1000)	
@	CHAR32	nvarchar(1000)	
@	CHAR33	nvarchar(1000)	
@	CHAR34	nvarchar(1000)	
@	CHAR35	nvarchar(1000)	
@	CHAR36	nvarchar(1000)	

	Name	Data type	Description
@	CHAR37	nvarchar(1000)	
@	CHAR38	nvarchar(1000)	
@	CHAR39	nvarchar(1000)	
@	CHAR40	nvarchar(1000)	
@	CHAR41	nvarchar(1000)	
@	CHAR42	nvarchar(1000)	
@	CHAR43	nvarchar(1000)	
@	CHAR44	nvarchar(1000)	
@	CHAR45	nvarchar(1000)	
@	CHAR46	nvarchar(1000)	
@	CHAR47	nvarchar(1000)	
@	CHAR48	nvarchar(1000)	
@	CHAR49	nvarchar(1000)	
@	CHAR50	nvarchar(1000)	
@	CHAR51	nvarchar(1000)	
@	CHAR52	nvarchar(1000)	
@	CHAR53	nvarchar(1000)	
@	CHAR54	nvarchar(1000)	
@	CHAR55	nvarchar(1000)	
@	CHAR56	nvarchar(1000)	
@	CHAR57	nvarchar(1000)	
@	CHAR58	nvarchar(1000)	
@	CHAR59	nvarchar(1000)	
@	CHAR60	nvarchar(1000)	
@	CHAR61	nvarchar(1000)	
@	CHAR62	nvarchar(1000)	
@	CHAR63	nvarchar(1000)	
@	CHAR64	nvarchar(1000)	
@	CHAR65	nvarchar(1000)	
@	CHAR66	nvarchar(1000)	
@	CHAR67	nvarchar(1000)	
@	CHAR68	nvarchar(1000)	
@	CHAR69	nvarchar(1000)	
@	CHAR70	nvarchar(1000)	
@	NUM01	numeric(18, 4)	
@	NUM02	numeric(18, 4)	
@	NUM03	numeric(18, 4)	
@	NUM04	numeric(18, 4)	
@	NUM05	numeric(18, 4)	
@	NUM06	numeric(18, 4)	

	Name	Data type	Description
@	NUM07	numeric(18, 4)	
@	NUM08	numeric(18, 4)	
@	NUM09	numeric(18, 4)	
@	NUM10	numeric(18, 4)	
@	NUM11	numeric(18, 4)	
@	NUM12	numeric(18, 4)	
@	NUM13	numeric(18, 4)	
@	NUM14	numeric(18, 4)	
@	NUM15	numeric(18, 4)	
@	NUM16	numeric(18, 4)	
@	NUM17	numeric(18, 4)	
@	NUM18	numeric(18, 4)	
@	NUM19	numeric(18, 4)	
@	NUM20	numeric(18, 4)	
@	NUM21	numeric(18, 4)	
@	NUM22	numeric(18, 4)	
@	NUM23	numeric(18, 4)	
@	NUM24	numeric(18, 4)	
@	NUM25	numeric(18, 4)	
@	NUM26	numeric(18, 4)	
@	NUM27	numeric(18, 4)	
@	NUM28	numeric(18, 4)	
@	NUM29	numeric(18, 4)	
@	NUM30	numeric(18, 4)	
@	NUM31	numeric(18, 4)	
@	NUM32	numeric(18, 4)	
@	NUM33	numeric(18, 4)	
@	NUM34	numeric(18, 4)	
@	NUM35	numeric(18, 4)	
@	NUM36	numeric(18, 4)	
@	NUM37	numeric(18, 4)	
@	NUM38	numeric(18, 4)	
@	NUM39	numeric(18, 4)	
@	NUM40	numeric(18, 4)	
@	NUM41	numeric(20, 4)	
@	NUM42	numeric(20, 4)	
@	NUM43	numeric(20, 4)	
@	NUM44	numeric(20, 4)	
@	NUM45	numeric(20, 4)	
@	NUM46	numeric(20, 4)	

	Name	Data type	Description
@	NUM47	numeric(20, 4)	
@	NUM48	numeric(20, 4)	
@	NUM49	numeric(20, 4)	
@	NUM50	numeric(20, 4)	
@	NUM51	numeric(20, 4)	
@	NUM52	numeric(20, 4)	
@	NUM53	numeric(20, 4)	
@	NUM54	numeric(20, 4)	
@	NUM55	numeric(20, 4)	
@	NUM56	numeric(20, 4)	
@	NUM57	numeric(20, 4)	
@	NUM58	numeric(20, 4)	
@	NUM59	numeric(20, 4)	
@	NUM60	numeric(20, 4)	
@	NUM61	numeric(20, 4)	
@	NUM62	numeric(20, 4)	
@	NUM63	numeric(20, 4)	
@	NUM64	numeric(20, 4)	
@	NUM65	numeric(20, 4)	
@	NUM66	numeric(20, 4)	
@	NUM67	numeric(20, 4)	
@	NUM68	numeric(20, 4)	
@	NUM69	numeric(20, 4)	
@	NUM70	numeric(20, 4)	

Uses

Name	
	IPC_UPDATELOGDEFINE
	IPCLOGTRANS

Script

```
CREATE procedure [dbo].[IPC_UPDATELOGDEFINE]
@IPCTRANSID NUMERIC(18,4)
,@CHAR01 nvarchar(1000)
,@CHAR02 nvarchar(1000)
,@CHAR03 nvarchar(1000)
,@CHAR04 nvarchar(1000)
,@CHAR05 nvarchar(1000)
,@CHAR06 nvarchar(1000)
,@CHAR07 nvarchar(1000)
,@CHAR08 nvarchar(1000)
,@CHAR09 nvarchar(1000)
,@CHAR10 nvarchar(1000)
,@CHAR11 nvarchar(1000)
,@CHAR12 nvarchar(1000)
,@CHAR13 nvarchar(1000)
,@CHAR14 nvarchar(1000)
,@CHAR15 nvarchar(1000)
,@CHAR16 nvarchar(1000)
,@CHAR17 nvarchar(1000)
,@CHAR18 nvarchar(1000)
,@CHAR19 nvarchar(1000)
,@CHAR20 nvarchar(1000)
```

,@CHAR21 nvarchar(1000)
,@CHAR22 nvarchar(1000)
,@CHAR23 nvarchar(1000)
,@CHAR24 nvarchar(1000)
 ,@CHAR25 nvarchar(1000)
 ,@CHAR26 NVARCHAR(1000),
 @CHAR27 NVARCHAR(1000),
 @CHAR28 NVARCHAR(1000),
 @CHAR29 NVARCHAR(1000),
 @CHAR30 NVARCHAR(1000),
 @CHAR31 NVARCHAR(1000),
 @CHAR32 NVARCHAR(1000),
 @CHAR33 NVARCHAR(1000),
 @CHAR34 NVARCHAR(1000),
 @CHAR35 NVARCHAR(1000),
 @CHAR36 NVARCHAR(1000),
 @CHAR37 NVARCHAR(1000),
 @CHAR38 NVARCHAR(1000),
 @CHAR39 NVARCHAR(1000),
 @CHAR40 NVARCHAR(1000),
 @CHAR41 NVARCHAR(1000),
 @CHAR42 NVARCHAR(1000),
 @CHAR43 NVARCHAR(1000),
 @CHAR44 NVARCHAR(1000),
 @CHAR45 NVARCHAR(1000),
 @CHAR46 NVARCHAR(1000),
 @CHAR47 NVARCHAR(1000),
 @CHAR48 NVARCHAR(1000),
 @CHAR49 NVARCHAR(1000),
 @CHAR50 NVARCHAR(1000),
 @CHAR51 NVARCHAR(1000),
 @CHAR52 NVARCHAR(1000),
 @CHAR53 NVARCHAR(1000),
 @CHAR54 NVARCHAR(1000),
 @CHAR55 NVARCHAR(1000),
 @CHAR56 NVARCHAR(1000),
 @CHAR57 NVARCHAR(1000),
 @CHAR58 NVARCHAR(1000),
 @CHAR59 NVARCHAR(1000),
 @CHAR60 NVARCHAR(1000),
 @CHAR61 NVARCHAR(1000),
 @CHAR62 NVARCHAR(1000),
 @CHAR63 NVARCHAR(1000),
 @CHAR64 NVARCHAR(1000),
 @CHAR65 NVARCHAR(1000),
 @CHAR66 NVARCHAR(1000),
 @CHAR67 NVARCHAR(1000),
 @CHAR68 NVARCHAR(1000),
 @CHAR69 NVARCHAR(1000),
 @CHAR70 NVARCHAR(1000)

,@NUM01 NUMERIC(18,4)
,@NUM02 NUMERIC(18,4)
,@NUM03 NUMERIC(18,4)
,@NUM04 NUMERIC(18,4)
,@NUM05 NUMERIC(18,4)
,@NUM06 NUMERIC(18,4)
,@NUM07 NUMERIC(18,4)
,@NUM08 NUMERIC(18,4)
,@NUM09 NUMERIC(18,4)
,@NUM10 NUMERIC(18,4)
,@NUM11 NUMERIC(18,4)
,@NUM12 NUMERIC(18,4)
,@NUM13 NUMERIC(18,4)
,@NUM14 NUMERIC(18,4)
,@NUM15 NUMERIC(18,4)
,@NUM16 NUMERIC(18,4)
,@NUM17 NUMERIC(18,4)
,@NUM18 NUMERIC(18,4)
,@NUM19 NUMERIC(18,4)
,@NUM20 NUMERIC(18,4)
 ,@NUM21 NUMERIC(18,4)
,@NUM22 NUMERIC(18,4)
,@NUM23 NUMERIC(18,4)
,@NUM24 NUMERIC(18,4)
,@NUM25 NUMERIC(18,4),
 @NUM26 NUMERIC(18,4),
 @NUM27 NUMERIC(18,4),
 @NUM28 NUMERIC(18,4),
 @NUM29 NUMERIC(18,4),
 @NUM30 NUMERIC(18,4),
 @NUM31 NUMERIC(18,4),
 @NUM32 NUMERIC(18,4),
 @NUM33 NUMERIC(18,4),
 @NUM34 NUMERIC(18,4),
 @NUM35 NUMERIC(18,4),
 @NUM36 NUMERIC(18,4),
 @NUM37 NUMERIC(18,4),
 @NUM38 NUMERIC(18,4),
 @NUM39 NUMERIC(18,4),
 @NUM40 NUMERIC(18,4),
 @NUM41 NUMERIC(20,4),
 @NUM42 NUMERIC(20,4),
 @NUM43 NUMERIC(20,4),

```

@NUM44      NUMERIC(20,4),
@NUM45      NUMERIC(20,4),
@NUM46      NUMERIC(20,4),
@NUM47      NUMERIC(20,4),
@NUM48      NUMERIC(20,4),
@NUM49      NUMERIC(20,4),
@NUM50      NUMERIC(20,4),
@NUM51      NUMERIC(20,4),
@NUM52      NUMERIC(20,4),
@NUM53      NUMERIC(20,4),
@NUM54      NUMERIC(20,4),
@NUM55      NUMERIC(20,4),
@NUM56      NUMERIC(20,4),
@NUM57      NUMERIC(20,4),
@NUM58      NUMERIC(20,4),
@NUM59      NUMERIC(20,4),
@NUM60      NUMERIC(20,4),
@NUM61      NUMERIC(20,4),
@NUM62      NUMERIC(20,4),
@NUM63      NUMERIC(20,4),
@NUM64      NUMERIC(20,4),
@NUM65      NUMERIC(20,4),
@NUM66      NUMERIC(20,4),
@NUM67      NUMERIC(20,4),
@NUM68      NUMERIC(20,4),
@NUM69      NUMERIC(20,4),
@NUM70      NUMERIC(20,4)
AS
BEGIN
--30.8.2017 minh modify char49
declare @topline as varchar(200)
--select @char49= CHAR49,@topline=SUBSTRING(CHAR49,1,case when CHARINDEX(CHAR(13),CHAR49,1)=0 then len(char49) else
CHARINDEX(CHAR(13),CHAR49,1) end ) from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID
select @topline=SUBSTRING(CHAR49,1,case when CHARINDEX(CHAR(13),CHAR49,1)=0 then len(char49) else
CHARINDEX(CHAR(13),CHAR49,1)-1 end ) from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID
declare @lenchar49 as int
select @lenchar49=len(char49)  from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID
if(@lenchar49>500) update IPCLOGTRANS set CHAR49=SUBSTRING(char49,1,500) where IPCTRANSID=@IPCTRANSID

UPDATE [dbo].[IPCLOGTRANS]
SET
[CHAR01] = CASE WHEN @CHAR01 = '' THEN CHAR01 ELSE @CHAR01 END
,[CHAR02] = CASE WHEN @CHAR02 = '' THEN CHAR02 ELSE @CHAR02 END
,[CHAR03] = CASE WHEN @CHAR03 = '' THEN CHAR03 ELSE @CHAR03 END
,[CHAR04] = CASE WHEN @CHAR04 = '' THEN CHAR04 ELSE @CHAR04 END
,[CHAR05] = CASE WHEN @CHAR05 = '' THEN CHAR05 ELSE @CHAR05 END
,[CHAR06] = CASE WHEN @CHAR06 = '' THEN CHAR06 ELSE @CHAR06 END
,[CHAR07] = CASE WHEN @CHAR07 = '' THEN CHAR07 ELSE @CHAR07 END
,[CHAR08] = CASE WHEN @CHAR08 = '' THEN CHAR08 ELSE @CHAR08 END
,[CHAR09] = CASE WHEN @CHAR09 = '' THEN CHAR09 ELSE @CHAR09 END
,[CHAR10] = CASE WHEN @CHAR10 = '' THEN CHAR10 ELSE @CHAR10 END
,[CHAR11] = CASE WHEN @CHAR11 = '' THEN CHAR11 ELSE @CHAR11 END
,[CHAR12] = CASE WHEN @CHAR12 = '' THEN CHAR12 ELSE @CHAR12 END
,[CHAR13] = CASE WHEN @CHAR13 = '' THEN CHAR13 ELSE @CHAR13 END
,[CHAR14] = CASE WHEN @CHAR14 = '' THEN CHAR14 ELSE @CHAR14 END
,[CHAR15] = CASE WHEN @CHAR15 = '' THEN CHAR15 ELSE @CHAR15 END
,[CHAR16] = CASE WHEN @CHAR16 = '' THEN CHAR16 ELSE @CHAR16 END
,[CHAR17] = CASE WHEN @CHAR17 = '' THEN CHAR17 ELSE @CHAR17 END
,[CHAR18] = CASE WHEN @CHAR18 = '' THEN CHAR18 ELSE @CHAR18 END
,[CHAR19] = CASE WHEN @CHAR19 = '' THEN CHAR19 ELSE @CHAR19 END
,[CHAR20] = CASE WHEN @CHAR20 = '' THEN CHAR20 ELSE @CHAR20 END
,[CHAR21] = CASE WHEN @CHAR21 = '' THEN CHAR21 ELSE @CHAR21 END
,[CHAR22] = CASE WHEN @CHAR22 = '' THEN CHAR22 ELSE @CHAR22 END
,[CHAR23] = CASE WHEN @CHAR23 = '' THEN CHAR23 ELSE @CHAR23 END
,[CHAR24] = CASE WHEN @CHAR24 = '' THEN CHAR24 ELSE @CHAR24 END
,[CHAR25] = CASE WHEN @CHAR25 = '' THEN CHAR25 ELSE @CHAR25 END
,[CHAR26] = CASE WHEN @CHAR26 = '' THEN CHAR26 ELSE @CHAR26 END
,[CHAR27] = CASE WHEN @CHAR27 = '' THEN CHAR27 ELSE @CHAR27 END
,[CHAR28] = CASE WHEN @CHAR28 = '' THEN CHAR28 ELSE @CHAR28 END
,[CHAR29] = CASE WHEN @CHAR29 = '' THEN CHAR29 ELSE @CHAR29 END
,[CHAR30] = CASE WHEN @CHAR30 = '' THEN CHAR30 ELSE @CHAR30 END
,[CHAR31] = CASE WHEN @CHAR31 = '' THEN CHAR31 ELSE @CHAR31 END
,[CHAR32] = CASE WHEN @CHAR32 = '' THEN CHAR32 ELSE @CHAR32 END
,[CHAR33] = CASE WHEN @CHAR33 = '' THEN CHAR33 ELSE @CHAR33 END
,[CHAR34] = CASE WHEN @CHAR34 = '' THEN CHAR34 ELSE @CHAR34 END
,[CHAR35] = CASE WHEN @CHAR35 = '' THEN CHAR35 ELSE @CHAR35 END
,[CHAR36] = CASE WHEN @CHAR36 = '' THEN CHAR36 ELSE @CHAR36 END
,[CHAR37] = CASE WHEN @CHAR37 = '' THEN CHAR37 ELSE @CHAR37 END
,[CHAR38] = CASE WHEN @CHAR38 = '' THEN CHAR38 ELSE @CHAR38 END
,[CHAR39] = CASE WHEN @CHAR39 = '' THEN CHAR39 ELSE @CHAR39 END
,[CHAR40] = CASE WHEN @CHAR40 = '' THEN CHAR40 ELSE @CHAR40 END
,[CHAR41] = CASE WHEN @CHAR41 = '' THEN CHAR41 ELSE @CHAR41 END
,[CHAR42] = CASE WHEN @CHAR42 = '' THEN CHAR42 ELSE @CHAR42 END
,[CHAR43] = CASE WHEN @CHAR43 = '' THEN CHAR43 ELSE @CHAR43 END
,[CHAR44] = CASE WHEN @CHAR44 = '' THEN CHAR44 ELSE @CHAR44 END
,[CHAR45] = CASE WHEN @CHAR45 = '' THEN CHAR45 ELSE @CHAR45 END
,[CHAR46] = CASE WHEN @CHAR46 = '' THEN CHAR46 ELSE @CHAR46 END
,[CHAR47] = CASE WHEN @CHAR47 = '' THEN CHAR47 ELSE @CHAR47 END
,[CHAR48] = CASE WHEN @CHAR48 = '' THEN CHAR48 ELSE @CHAR48 END
,[CHAR49] = CASE WHEN @CHAR49 = '' THEN CHAR49 ELSE CASE WHEN @CHAR49=@topline THEN CHAR49 ELSE case when @topline=''
then @CHAR49 else @CHAR49 + CHAR(13) + CHAR49 end END END
,[CHAR50] = CASE WHEN @CHAR50 = '' THEN CHAR50 ELSE @CHAR50 END
,[CHAR51] = CASE WHEN @CHAR51 = '' THEN CHAR51 ELSE @CHAR51 END

```

```

,[CHAR52] = CASE WHEN @CHAR52 = '' THEN CHAR52 ELSE @CHAR52 END
,[CHAR53] = CASE WHEN @CHAR53 = '' THEN CHAR53 ELSE @CHAR53 END
,[CHAR54] = CASE WHEN @CHAR54 = '' THEN CHAR54 ELSE @CHAR54 END
,[CHAR55] = CASE WHEN @CHAR55 = '' THEN CHAR55 ELSE @CHAR55 END
,[CHAR56] = CASE WHEN @CHAR56 = '' THEN CHAR56 ELSE @CHAR56 END
,[CHAR57] = CASE WHEN @CHAR57 = '' THEN CHAR57 ELSE @CHAR57 END
,[CHAR58] = CASE WHEN @CHAR58 = '' THEN CHAR58 ELSE @CHAR58 END
,[CHAR59] = CASE WHEN @CHAR59 = '' THEN CHAR59 ELSE @CHAR59 END
,[CHAR60] = CASE WHEN @CHAR60 = '' THEN CHAR60 ELSE @CHAR60 END
,[CHAR61] = CASE WHEN @CHAR61 = '' THEN CHAR61 ELSE @CHAR61 END
,[CHAR62] = CASE WHEN @CHAR62 = '' THEN CHAR62 ELSE @CHAR62 END
,[CHAR63] = CASE WHEN @CHAR63 = '' THEN CHAR63 ELSE @CHAR63 END
,[CHAR64] = CASE WHEN @CHAR64 = '' THEN CHAR64 ELSE @CHAR64 END
,[CHAR65] = CASE WHEN @CHAR65 = '' THEN CHAR65 ELSE @CHAR65 END
,[CHAR66] = CASE WHEN @CHAR66 = '' THEN CHAR66 ELSE @CHAR66 END
,[CHAR67] = CASE WHEN @CHAR67 = '' THEN CHAR67 ELSE @CHAR67 END
,[CHAR68] = CASE WHEN @CHAR68 = '' THEN CHAR68 ELSE @CHAR68 END
,[CHAR69] = CASE WHEN @CHAR69 = '' THEN CHAR69 ELSE @CHAR69 END
,[CHAR70] = CASE WHEN @CHAR70 = '' THEN CHAR70 ELSE @CHAR70 END

,[NUM01] = CASE WHEN @NUM01 = 0 THEN NUM01 ELSE @NUM01 END
,[NUM02] = CASE WHEN @NUM02 = 0 THEN NUM02 ELSE @NUM02 END
,[NUM03] = CASE WHEN @NUM03 = 0 THEN NUM03 ELSE @NUM03 END
,[NUM04] = CASE WHEN @NUM04 = 0 THEN NUM04 ELSE @NUM04 END
,[NUM05] = CASE WHEN @NUM05 = 0 THEN NUM05 ELSE @NUM05 END
,[NUM06] = CASE WHEN @NUM06 = 0 THEN NUM06 ELSE @NUM06 END
,[NUM07] = CASE WHEN @NUM07 = 0 THEN NUM07 ELSE @NUM07 END
,[NUM08] = CASE WHEN @NUM08 = 0 THEN NUM08 ELSE @NUM08 END
,[NUM09] = CASE WHEN @NUM09 = 0 THEN NUM09 ELSE @NUM09 END
,[NUM10] = CASE WHEN @NUM10 = 0 THEN NUM10 ELSE @NUM10 END
,[NUM11] = CASE WHEN @NUM11 = 0 THEN NUM11 ELSE @NUM11 END
,[NUM12] = CASE WHEN @NUM12 = 0 THEN NUM12 ELSE @NUM12 END
,[NUM13] = CASE WHEN @NUM13 = 0 THEN NUM13 ELSE @NUM13 END
,[NUM14] = CASE WHEN @NUM14 = 0 THEN NUM14 ELSE @NUM14 END
,[NUM15] = CASE WHEN @NUM15 = 0 THEN NUM15 ELSE @NUM15 END
,[NUM16] = CASE WHEN @NUM16 = 0 THEN NUM16 ELSE @NUM16 END
,[NUM17] = CASE WHEN @NUM17 = 0 THEN NUM17 ELSE @NUM17 END
,[NUM18] = CASE WHEN @NUM18 = 0 THEN NUM18 ELSE @NUM18 END
,[NUM19] = CASE WHEN @NUM19 = 0 THEN NUM19 ELSE @NUM19 END
,[NUM20] = CASE WHEN @NUM20 = 0 THEN NUM20 ELSE @NUM20 END
,[NUM21] = CASE WHEN @NUM21 = 0 THEN NUM21 ELSE @NUM21 END
,[NUM22] = CASE WHEN @NUM22 = 0 THEN NUM22 ELSE @NUM22 END
,[NUM23] = CASE WHEN @NUM23 = 0 THEN NUM23 ELSE @NUM23 END
,[NUM24] = CASE WHEN @NUM24 = 0 THEN NUM24 ELSE @NUM24 END
,[NUM25] = CASE WHEN @NUM25 = 0 THEN NUM25 ELSE @NUM25 END
,[NUM26] = CASE WHEN @NUM26 = 0 THEN NUM26 ELSE @NUM26 END
,[NUM27] = CASE WHEN @NUM27 = 0 THEN NUM27 ELSE @NUM27 END
,[NUM28] = CASE WHEN @NUM28 = 0 THEN NUM28 ELSE @NUM28 END
,[NUM29] = CASE WHEN @NUM29 = 0 THEN NUM29 ELSE @NUM29 END
,[NUM30] = CASE WHEN @NUM30 = 0 THEN NUM30 ELSE @NUM30 END
,[NUM31] = CASE WHEN @NUM31 = 0 THEN NUM31 ELSE @NUM31 END
,[NUM32] = CASE WHEN @NUM32 = 0 THEN NUM32 ELSE @NUM32 END
,[NUM33] = CASE WHEN @NUM33 = 0 THEN NUM33 ELSE @NUM33 END
,[NUM34] = CASE WHEN @NUM34 = 0 THEN NUM34 ELSE @NUM34 END
,[NUM35] = CASE WHEN @NUM35 = 0 THEN NUM35 ELSE @NUM35 END
,[NUM36] = CASE WHEN @NUM36 = 0 THEN NUM36 ELSE @NUM36 END
,[NUM37] = CASE WHEN @NUM37 = 0 THEN NUM37 ELSE @NUM37 END
,[NUM38] = CASE WHEN @NUM38 = 0 THEN NUM38 ELSE @NUM38 END
,[NUM39] = CASE WHEN @NUM39 = 0 THEN NUM39 ELSE @NUM39 END
,[NUM40] = CASE WHEN @NUM40 = 0 THEN NUM40 ELSE @NUM40 END
,[NUM41] = CASE WHEN @NUM41 = 0 THEN NUM41 ELSE @NUM41 END
,[NUM42] = CASE WHEN @NUM42 = 0 THEN NUM42 ELSE @NUM42 END
,[NUM43] = CASE WHEN @NUM43 = 0 THEN NUM43 ELSE @NUM43 END
,[NUM44] = CASE WHEN @NUM44 = 0 THEN NUM44 ELSE @NUM44 END
,[NUM45] = CASE WHEN @NUM45 = 0 THEN NUM45 ELSE @NUM45 END
,[NUM46] = CASE WHEN @NUM46 = 0 THEN NUM46 ELSE @NUM46 END
,[NUM47] = CASE WHEN @NUM47 = 0 THEN NUM47 ELSE @NUM47 END
,[NUM48] = CASE WHEN @NUM48 = 0 THEN NUM48 ELSE @NUM48 END
,[NUM49] = CASE WHEN @NUM49 = 0 THEN NUM49 ELSE @NUM49 END
,[NUM50] = CASE WHEN @NUM50 = 0 THEN NUM50 ELSE @NUM50 END
,[NUM51] = CASE WHEN @NUM51 = 0 THEN NUM51 ELSE @NUM51 END
,[NUM52] = CASE WHEN @NUM52 = 0 THEN NUM52 ELSE @NUM52 END
,[NUM53] = CASE WHEN @NUM53 = 0 THEN NUM53 ELSE @NUM53 END
,[NUM54] = CASE WHEN @NUM54 = 0 THEN NUM54 ELSE @NUM54 END
,[NUM55] = CASE WHEN @NUM55 = 0 THEN NUM55 ELSE @NUM55 END
,[NUM56] = CASE WHEN @NUM56 = 0 THEN NUM56 ELSE @NUM56 END
,[NUM57] = CASE WHEN @NUM57 = 0 THEN NUM57 ELSE @NUM57 END
,[NUM58] = CASE WHEN @NUM58 = 0 THEN NUM58 ELSE @NUM58 END
,[NUM59] = CASE WHEN @NUM59 = 0 THEN NUM59 ELSE @NUM59 END
,[NUM60] = CASE WHEN @NUM60 = 0 THEN NUM60 ELSE @NUM60 END
,[NUM61] = CASE WHEN @NUM61 = 0 THEN NUM61 ELSE @NUM61 END
,[NUM62] = CASE WHEN @NUM62 = 0 THEN NUM62 ELSE @NUM62 END
,[NUM63] = CASE WHEN @NUM63 = 0 THEN NUM63 ELSE @NUM63 END
,[NUM64] = CASE WHEN @NUM64 = 0 THEN NUM64 ELSE @NUM64 END
,[NUM65] = CASE WHEN @NUM65 = 0 THEN NUM65 ELSE @NUM65 END
,[NUM66] = CASE WHEN @NUM66 = 0 THEN NUM66 ELSE @NUM66 END
,[NUM67] = CASE WHEN @NUM67 = 0 THEN NUM67 ELSE @NUM67 END
,[NUM68] = CASE WHEN @NUM68 = 0 THEN NUM68 ELSE @NUM68 END
,[NUM69] = CASE WHEN @NUM69 = 0 THEN NUM69 ELSE @NUM69 END
,[NUM70] = CASE WHEN @NUM70 = 0 THEN NUM70 ELSE @NUM70 END

WHERE IPCTRANSID = @IPCTRANSID
END

```

5.2.25. Procedure: IPC_UPDATELOGTRANSError

Input/Output

Name		Data type	Description
@IPCTRANSID		varchar(20)	
@ERROR		varchar(1000)	

Uses

Name
IPC_UPDATELOGTRANSError
IPCLOGTRANS

Script

```
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[IPC_UPDATELOGTRANSError]
    -- Add the parameters for the stored procedure here
    @IPCTRANSID VARCHAR(20),
    @ERROR VARCHAR(1000)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    declare @topline as varchar(200)
--select @char49= CHAR49,@topline=SUBSTRING(CHAR(13),CHAR49,1)=0 then len(char49) else
CHARINDEX(CHAR(13),CHAR49,1) end ) from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID
select @topline=SUBSTRING(CHAR(13),CHAR49,1)=0 then len(char49) else
CHARINDEX(CHAR(13),CHAR49,1)-1 end ) from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID
    UPDATE IPCLOGTRANS SET CHAR49=case when 'FAST-' + @ERROR=@topline then CHAR49
else
case when @topline='' then 'FAST-' + @ERROR else 'FAST-' + @ERROR + CHAR(13) + CHAR49 end
end
WHERE IPCTRANSID=@IPCTRANSID
END
```

5.2.26. Procedure: IPCAPPROVETRAN_TELLER

Input/Output

	Name	Data type	Description
→@	IPCTRANSID	numeric(18, 0)	
→@	USERID	varchar(50)	

Uses

	Name
⚙️	IPCAPPROVETRAN_TELLER
📄	IPCLOGTRANAPPROVE
📄	IPCLOGTRANS

TRIAL

Script

```
CREATE procedure [dbo].[IPCAPPROVETRAN_TELLER]
    -- Add the parameters for the stored procedure here
    @IPCTRANSID NUMERIC, @USERID VARCHAR(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    DECLARE @USERTYPE VARCHAR(10)
    DECLARE @USERLEV INT
    DECLARE @PRODUCTID VARCHAR(50)
    DECLARE @USERDOTRAN VARCHAR(50)
    DECLARE @WORKDATE DATETIME
    DECLARE @NEXTLEVAPP INT
    DECLARE @USERNAME NVARCHAR(255)
    SET @NEXTLEVAPP = -1
    SELECT @USERTYPE = USERTYPE, @USERLEV = USERLEVEL, @USERNAME = FULLNAME
        FROM EBA_USERS WHERE UserID = @USERID
    DECLARE @TRANCODE VARCHAR(50)
    DECLARE @AMOUNT NUMERIC
    DECLARE @CCYID VARCHAR(10)
    SELECT TOP 1 @TRANCODE = IPCTRANCODE, @AMOUNT = NUM01, @CCYID = CCYID, @WORKDATE = IPCWORKDATE, @USERDOTRAN =
    @USERID
        FROM IPCLOGTRANS
        WHERE IPCTRANSID = @IPCTRANSID AND (NEXTUSERAPP = @USERLEV)
    SELECT @PRODUCTID = PRODUCTID FROM EBA_CONTRACT
        WHERE CONTRACTNO IN (SELECT CONTRACTNO FROM EBA_Users WHERE UserID = @USERDOTRAN)

    IF NOT EXISTS (SELECT * FROM IPCLOGTRANAPPROVE WHERE IPCTRANSID = @IPCTRANSID AND USERAPP = @USERID )
    BEGIN
        INSERT INTO IPCLOGTRANAPPROVE VALUES(@IPCTRANSID, GETDATE(), @WORKDATE, @USERID, '', '')
    END

    DECLARE @FLAG INT
    SET @FLAG = -1

    DECLARE @CURORD INT
    SET @CURORD = -1
    SELECT @CURORD = ORDAPP FROM IBS_TELLERAPPTRANDETAIL
        WHERE APPTRANID IN (SELECT APPTRANID FROM IBS_TELLERAPPTRAN
            WHERE TRANCODE = @TRANCODE AND PRODUCTID = @PRODUCTID AND
            CCYID = @CCYID
                AND USERLEVEL = @USERLEV
                    AND @AMOUNT BETWEEN FROMLIMIT AND TOLIMIT)

    SELECT TOP 1 @NEXTLEVAPP = USERLEVEL FROM IBS_TELLERAPPTRANDETAIL
        WHERE APPTRANID IN (SELECT APPTRANID FROM IBS_TELLERAPPTRAN
            WHERE TRANCODE = @TRANCODE AND PRODUCTID = @PRODUCTID AND
            CCYID = @CCYID
                AND @AMOUNT BETWEEN FROMLIMIT AND TOLIMIT AND ORDAPP >
            @CURORD)
            ORDER BY ORDAPP
            IF @NEXTLEVAPP = -1
                BEGIN
                    UPDATE IPCLOGTRANS SET
                        NEXTUSERAPP = '',
                        LISTUSERAPP = LISTUSERAPP + '#' + @USERID,
                        USERCURAPP = @USERID,
                        APPRSTS = '3'
                    WHERE IPCTRANSID = @IPCTRANSID OR (ISBATCH = 'Y' AND
                        BATCHREF = @IPCTRANSID)

                    SELECT 1 EXECTRAN
                    RETURN
                END
            ELSE
                BEGIN
                    UPDATE IPCLOGTRANS SET
                        NEXTUSERAPP = @NEXTLEVAPP,
                        LISTUSERAPP = LISTUSERAPP + '#' + @USERID,
                        USERCURAPP = @USERID,
                        APPRSTS = '2'
                    WHERE IPCTRANSID = @IPCTRANSID OR (ISBATCH = 'Y' AND
                        BATCHREF = @IPCTRANSID)

                    SELECT 0 EXECTRAN
                    RETURN
                END
    END
```

5.2.27. Procedure: IPCAPPROVETRAN_USER

Input/Output

	Name	Data type	Description
@	IPCTRANSID	numeric(18, 0)	
@	USERID	varchar(50)	
@	AUTHENTYPE	varchar(50)	
@	AUTHENCODE	varchar(50)	

Uses

Name
IPCAPPROVETRAN_USER
IPCLOGTRANAPPROVE
IPCLOGTRANS

Script

```

CREATE procedure [dbo].[IPCAPPROVETRAN_USER]
    -- Add the parameters for the stored procedure here
    -- @IPCTRANSID NUMERIC, @USERID VARCHAR(50), @AUTHENTYPE VARCHAR(50), @AUTHENCODE VARCHAR(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SET NOCOUNT ON;
    DECLARE @USERTYPE VARCHAR(10)
    DECLARE @USERLEV INT
    DECLARE @CONTRACTNO VARCHAR(50)
    DECLARE @WORKDATE DATETIME
    DECLARE @NEXTLEVAPP INT
    DECLARE @USERNAME NVARCHAR(255)
    SET @NEXTLEVAPP = -1
    SELECT @USERTYPE = USERTYPE, @USERLEV = USERLEVEL, @CONTRACTNO = CONTRACTNO, @USERNAME = FULLNAME
        FROM EBA_USERS WHERE USERID = @USERID
    DECLARE @TRANCODE VARCHAR(50)
    DECLARE @AMOUNT NUMERIC
    DECLARE @CCYID VARCHAR(10)
    SELECT @TRANCODE = IPCTRANCODE, @AMOUNT = NUM01, @CCYID = CCYID, @WORKDATE = IPCWORKDATE FROM IPCLOGTRANS
        WHERE IPCTRANSID = @IPCTRANSID AND NEXTUSERAPP = @USERLEV

    INSERT INTO IPCLOGTRANAPPROVE VALUES (@IPCTRANSID, GETDATE(), @WORKDATE, @USERID, @AUTHENTYPE, @AUTHENCODE)
    DECLARE @FLAG INT
    SET @FLAG = -1

    IF (@USERLEV = 2)
        BEGIN
            UPDATE IPCLOGTRANS SET
                NEXTUSERAPP = '1',
                LISTUSERAPP = LISTUSERAPP + '#' + @USERID,
                USERCURAPP = @USERID
            WHERE IPCTRANSID = @IPCTRANSID OR (ISBATCH = 'Y' AND BATCHREF = @IPCTRANSID)
            SELECT 0 EXECTRAN
            RETURN
        END
    IF (@USERLEV = 1)
        BEGIN
            DECLARE @PRODUCTID VARCHAR(50)
            SELECT @PRODUCTID = PRODUCTID FROM EBA_CONTRACT WHERE CONTRACTNO = @CONTRACTNO
            SELECT TOP 1 @NEXTLEVAPP = USERLEVEL FROM IBS_TELLERAPPTRANDETAIL
            WHERE APPTRANID IN (SELECT APPTRANID FROM IBS_TELLERAPPTRAN WHERE
                TRANCODE = @TRANCODE AND CCYID = @CCYID AND PRODUCTID = @PRODUCTID
                AND @AMOUNT BETWEEN FROMLIMIT AND TOLIMIT
                ORDER BY ORDAPP)
            IF (@NEXTLEVAPP <> -1)
                BEGIN
                    UPDATE IPCLOGTRANS SET
                        NEXTUSERAPP = @NEXTLEVAPP,
                        LISTUSERAPP = LISTUSERAPP + '#' + @USERID,
                        USERCURAPP = @USERID,
                        APPRSTS = '2'
                    WHERE IPCTRANSID = @IPCTRANSID OR (ISBATCH = 'Y' AND
                        
```

```

BATCHREF = @IPCTRANSID)
SELECT 0 EXECTRAN
RETURN
END
ELSE
BEGIN
UPDATE IPCLOGTRANS SET
NEXTUSERAPP = '',
LISTUSERAPP = LISTUSERAPP + '#' + @USERID,
USERCURAPP = @USERID,
APPRSTS = '3'
WHERE IPCTRANSID = @IPCTRANSID OR (ISBATCH = 'Y' AND
BATCHREF = @IPCTRANSID)
SELECT 1 EXECTRAN
RETURN
END
END

IF(@USERLEV NOT IN ('1','2'))
BEGIN
DECLARE @CURORD INT
SET @CURORD = -1
SELECT @CURORD = ORDAPP FROM IBS_USERAPPTRANDETAIL
WHERE APPTRANID IN (SELECT APPTRANID FROM IBS_USERAPPTRAN
WHERE TRANCODE = @TRANCODE AND CONTRACTNO = @CONTRACTNO
AND CCYID = @CCYID AND (CAST(@AMOUNT AS NUMERIC(18,3)) BETWEEN
FROMLIMIT AND TOLIMIT))
AND USERLEVEL = @USERLEV

SELECT TOP 1 @NEXTLEVAPP = USERLEVEL FROM IBS_USERAPPTRANDETAIL
WHERE APPTRANID IN (SELECT APPTRANID FROM IBS_USERAPPTRAN
WHERE TRANCODE = @TRANCODE AND CONTRACTNO = @CONTRACTNO
AND CCYID = @CCYID AND (CAST(@AMOUNT AS NUMERIC(18,3)) BETWEEN
FROMLIMIT AND TOLIMIT)
ORDER BY ORDAPP
IF @NEXTLEVAPP = -1
BEGIN
IF EXISTS (SELECT * FROM EBA_USERS WHERE CONTRACTNO = @CONTRACTNO AND USERLEVEL =
2)
SET @NEXTLEVAPP = 2
ELSE
SET @NEXTLEVAPP = 1
--SET @NEXTLEVAPP = 2
END
UPDATE IPCLOGTRANS SET
NEXTUSERAPP = @NEXTLEVAPP,
LISTUSERAPP = LISTUSERAPP + '#' + @USERID,
USERCURAPP = @USERID
WHERE IPCTRANSID = @IPCTRANSID OR (ISBATCH = 'Y' AND BATCHREF = @IPCTRANSID)
SELECT 0 EXECTRAN
RETURN
END
--0:CUSTSITE PASS, 1: WAITTING CUSTSITE, 2: WAITTING BANKSITE, 3: PASS ALL
END

```

5.2.28. Procedure: IPCCHECKEXIST_TRANREF

Input/Output

	Name	Data type	Description
→@	SOURCEID	varchar(20)	
→@	DESTID	varchar(20)	
→@	IPCTRANCODE	varchar(20)	
→@	SOURCETRANREF	varchar(50)	

Uses

Name
IPCHECKEXIST_TRANREF
IPCLOGTRANS

Script

```
-- =====
-- Author: Admin
-- Create date: 14/02/2009
-- Description: Check Exist Source & Dest TranRef
-- =====
CREATE procedure [dbo].[IPCCHECKEXIST_TRANREF]
    @SOURCEID           VARCHAR(20),
    @DESTID             VARCHAR(20),
    @IPCTRANCODE        VARCHAR(20),
    @SOURCETRANREF      VARCHAR(50)

AS
BEGIN
    SELECT * FROM IPCLOGTRANS A (NOLOCK)
    WHERE A.SOURCEID = @SOURCEID AND A.DESTID = @DESTID
        --AND A.IPCTRANCODE = @IPCTRANCODE
        --AND A.SOURCETRANREF = @SOURCETRANREF
        AND A.STATUS = 1
END
```

5.2.29. Procedure: IPCCOMPONENT_SELECT

Input/Output

	Name	Data type	Description
→@	SOURCEID	varchar(20)	
→@	DESTID	varchar(20)	
→@	IPCTRANCODE	varchar(20)	
→@	REVERSAL	varchar(1)	

Uses

	Name
⚙️	IPCCOMPONENT_SELECT
📄	IPCCOMPONENT (IPC component)
📄	IPCSYSVAR

Script

```
-- =====
-- Author: Admin
-- Create date: 14/02/2009
-- Description: Select Component Info
-- =====
CREATE procedure [dbo].[IPCCOMPONENT_SELECT]
    @SOURCEID           VARCHAR(20),
    @DESTID             VARCHAR(20),
    @IPCTRANCODE        VARCHAR(20),
    @REVERSAL           VARCHAR(1)
AS
BEGIN
    DECLARE @SYSTEMSTATUS VARCHAR(1)
    SET @SYSTEMSTATUS = '0'
    --SELECT @SYSTEMSTATUS = CASE WHEN A.VARVALUE = '0' THEN ISNULL(B.LINESTATUS, A.VARVALUE) ELSE A.VARVALUE END
    SELECT @SYSTEMSTATUS = CASE WHEN A.VARVALUE = '0' THEN A.VARVALUE ELSE A.VARVALUE END
    FROM IPCSYSVAR A (NOLOCK)
    --LEFT OUTER JOIN SYSCOMPANY B (NOLOCK) ON B.COMPID = @DESTID
    WHERE A.VARNAME = 'SYSTEMSTATUS'
    --
    SELECT A.ASSEMBLYFILE, A.ASSEMBLYTITLE, A.METHOD, A.PARMLIST, A.STATUS, A.ORD, A.SU, A.FA
    FROM IPCCOMPONENT A (NOLOCK)
    WHERE A.SOURCEID = @SOURCEID AND A.DESTID = @DESTID AND A.IPCTRANCODE = @IPCTRANCODE
        AND A.REVERSAL = CASE WHEN @REVERSAL = 'N' THEN 0 ELSE 1 END
        AND A.STATUS = @SYSTEMSTATUS
    ORDER BY A.ORD
END
```

5.2.30. Procedure: IPCCOMPONENTSYN_SELECT

Input/Output

Name		Data type	Description
@SOURCEID	SOURCEID	varchar(20)	
@DESTID	DESTID	varchar(20)	
@IPCTRANCODE	IPCTRANCODE	varchar(20)	

Uses

Name
IPCCOMPONENTSYN_SELECT
IPCCOMPONENT (IPC component)

Script

```
-- =====
-- Author: Admin
-- Create date: 14/02/2009
-- Description: Select Component Info To Synchronize Offline Transaction
-- =====
CREATE procedure [dbo].[IPCCOMPONENTSYN_SELECT]
    @SOURCEID           VARCHAR(20),
    @DESTID             VARCHAR(20),
    @IPCTRANCODE        VARCHAR(20)
AS
BEGIN
    SELECT A.ASSEMBLYFILE, A.ASSEMBLYTITLE, A.METHOD, A.PARMLIST
    FROM IPCCOMPONENT A (NOLOCK) --, IPCSYSVAR B (NOLOCK)
    WHERE A.SOURCEID = @SOURCEID AND A.DESTID = @DESTID AND A.IPCTRANCODE = @IPCTRANCODE
        AND A.STATUS = 3 AND A.REVERSAL = 0
        --AND B.VARNAME = 'SYSTEMSTATUS' AND B.VARVALUE = '3'
        --AND A.METHOD NOT IN ('CheckReversal')
    ORDER BY A.ORD
END
```

5.2.31. Procedure: IPCGETMAPPINGCOLNAME

Input/Output

	Name	Data type	Description
→@	IPCTRANCODE	varchar(50)	
→@	DESTID	varchar(50)	

Uses

Name
IPCGETMAPPINGCOLNAME
IPCMAPPINGCOLNAME

Script

```
CREATE procedure [dbo].[IPCGETMAPPINGCOLNAME]
    -- Add the parameters for the stored procedure here
    @IPCTRANCODE VARCHAR(50), @DESTID VARCHAR(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    -- Insert statements for procedure here
    SELECT * FROM IPCMAPPINGCOLNAME i
    WHERE i.IPCTRANCODE = @IPCTRANCODE AND i.DESTID = @DESTID
    ORDER BY i.ORD
END
```

5.2.32. Procedure: IPCGETSEQVALUE

Input/Output

Name		Data type	Description
@SEQNAME	SEQNAME	varchar(50)	
*@SEQVALUE	SEQVALUE	numeric(18, 0)	

Uses

Name
IPCGETSEQVALUE
IPCVINVENTORY

Script

```
-- =====
-- Author:          Admin
-- Create date:    14/02/2009
-- Description:    Get SEQ Value
-- =====
CREATE procedure [dbo].[IPCGETSEQVALUE]
(
    @SEQNAME VARCHAR(50),
    @SEQVALUE NUMERIC(18, 0) OUTPUT
)
AS
BEGIN
    UPDATE IPCINVENTORY SET @SEQVALUE = SEQVALUE,
                           SEQVALUE = CASE WHEN SEQVALUE = SEQMAX THEN 1 ELSE SEQVALUE + 1 END
    WHERE SEQNAME = @SEQNAME
END
```

5.2.33. Procedure: IPCGETSEQVALUE_SCALAR

Input/Output

Name	Data type	Description
@SEQNAME	varchar(50)	

Uses

Name
IPCGETSEQVALUE_SCALAR
IPCVINVENTORY

Script

```
-- =====
-- Author:          Admin
-- Create date: 14/02/2009
-- Description:    <GETSEQVALUE_SCALAR>
-- =====
CREATE procedure [dbo].[IPCGETSEQVALUE_SCALAR]
(
    @SEQNAME VARCHAR(50)
)
AS
BEGIN
    DECLARE @SEQVALUE AS NUMERIC(18, 0)
    UPDATE IPCINVENTORY SET @SEQVALUE = SEQVALUE,
        SEQVALUE = CASE WHEN SEQVALUE = SEQMAX THEN 1 ELSE SEQVALUE + 1 END
    WHERE SEQNAME = @SEQNAME

    SELECT @SEQVALUE SEQVALUE
END
```

5.2.34. Procedure: IPCLOGMESSAGE_INSERT

Input/Output

Name		Data type	Description
@IPCWORKDATE		varchar(10)	
@IPCTRANSID		numeric(18, 0)	
@IPCLOGTYPE		char(2)	
@MESSAGETYPE		varchar(10)	
@MESSAGE		nvarchar(4000)	

Uses

Name
IPCLOGMESSAGE_INSERT
IPCLOGMESSAGE

Script

```
-- =====
-- Author:          Admin
-- Create date: 14/02/2009
-- Description:    Log Transaction Message
-- =====
CREATE procedure [dbo].[IPCLOGMESSAGE_INSERT]
    @IPCWORKDATE      VARCHAR(10),
    @IPCTRANSID       NUMERIC(18, 0),
    @IPCLOGTYPE       CHAR(2),
    @MESSAGETYPE      VARCHAR(10),
    @MESSAGE          NVARCHAR(4000)
AS
BEGIN
    SET DATEFORMAT DMY
    INSERT INTO IPCLOGMESSAGE(IPCWORKDATE, IPCTRANSID, IPCLOGTYPE, MESSAGETYPE, MESSAGE)
    VALUES (@IPCWORKDATE, @IPCTRANSID, UPPER(@IPCLOGTYPE), @MESSAGETYPE, @MESSAGE)
END
```

5.2.35. Procedure: IPCLOGTRANAPPDETAIL_INSERT

Input/Output

	Name	Data type	Description
→@	IPCTRANSID	varchar(50)	
→@	IPCWORKDATE	varchar(50)	
→@	VALUE	nvarchar(MAX)	
→@	STATUS	varchar(1)	

Uses

Name
IPCLOGTRANAPPDETAIL_INSERT
IPCLOGTRANAPPDETAIL

Script

```
CREATE procedure [dbo].[IPCLOGTRANAPPDETAIL_INSERT]
    -- Add the parameters for the stored procedure here
    @IPCTRANSID VARCHAR(50), @IPCWORKDATE VARCHAR(50),
    @VALUE NVARCHAR(MAX), @STATUS VARCHAR(1)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SET DATEFORMAT DMY
    INSERT INTO IPCLOGTRANAPPDETAIL VALUES (@IPCTRANSID, @IPCWORKDATE, GETDATE(), @VALUE, @STATUS)
END
```

5.2.36. Procedure: IPCLOGTRANS_GETTRANSSYNC09

Input/Output

Name		Data type	Description
→@	TRANCODE	varchar(255)	
→@	STATUS	varchar(255)	

Uses

Name
IPCLOGTRANS_GETTRANSSYNC09
IPCSYSVAR

Script

```

CREATE procedure [dbo].[IPCLOGTRANS_GETTRANSSYNC09]
    @TRANCODE           VARCHAR (255),
    @STATUS              VARCHAR (255)
AS
BEGIN
    DECLARE @SCRIPT NVARCHAR(4000) = 'SELECT IPCTRANSID AS
PARENTREF,DESTERRORCODE,CCYID,STATUS,ONLINE,CHAR01,CHAR02,CHAR03,CHAR04,CHAR05,CHAR06,CHAR07,CHAR08,CHAR09,CHAR10,CHAR11,CHAR
12,CHAR13,CHAR14,CHAR15,CHAR16,CHAR17,CHAR18,CHAR19,CHAR20,CHAR21,CHAR22,CHAR23,CHAR24,CHAR25,CHAR26,CHAR27,CHAR28,CHAR29,CHA
R30,CHAR31,CHAR32,CHAR33,CHAR34,CHAR35,CHAR36,CHAR37,CHAR38,CHAR39,CHAR40,CHAR41,CHAR42,CHAR43,CHAR44,CHAR45,CHAR46,CHAR47,CH
AR48,CHAR49,CHAR50,CHAR51,CHAR52,CHAR53,CHAR54,CHAR55,CHAR56,CHAR57,CHAR58,CHAR59,CHAR60,CHAR61,CHAR62,CHAR63,CHAR64,CHAR65,C
HAR66,CHAR67,CHAR68,CHAR69,CHAR70,NUM01,NUM02,NUM03,NUM04,NUM05,NUM06,NUM07,NUM08,NUM09,NUM10,NUM11,NUM12,NUM13,NUM14,NUM15,N
UM16,NUM17,NUM18,NUM19,NUM20,NUM21,NUM22,NUM23,NUM24,NUM25,NUM26,NUM27,NUM28,NUM29,NUM30,NUM31,NUM32,NUM33,NUM34,NUM35,NUM36,
NUM37,NUM38,NUM39,NUM40,NUM41,NUM42,NUM43,NUM44,NUM45,NUM46,NUM47,NUM48,NUM49,NUM50,NUM51,NUM52,NUM53,NUM54,NUM55,NUM56,NUM57
,NUM58,NUM59,NUM60,NUM61,NUM62,NUM63,NUM64,NUM65,NUM66,NUM67,NUM68,NUM69,NUM70' FROM IPCLOGTRANS WHERE '
    DECLARE @TRETRY VARCHAR(20) = (SELECT VARVALUE FROM dbo.IPCSYSVAR WHERE VARNAME='TRETRY09')
    IF (@TRANCODE='')
        BEGIN
            SET @SCRIPT = @SCRIPT + ' STATUS IN ' + @STATUS + ' AND NUM69 < ' + @TRETRY + 'AND NUM70 = 1'
        END
    ELSE
        BEGIN
            SET @SCRIPT = @SCRIPT + ' STATUS IN ' + @STATUS + ' AND IPCTRANCODE IN ' + @TRANCODE + ' AND NUM69 < '
            + @TRETRY + 'AND NUM70 = 1'
        END
    EXEC SP_EXECUTE SQL @SCRIPT
END

```

5.2.37. Procedure: IPCLOGTRANS_GETTRANSSYNCREFUND

Input/Output

	Name	Data type	Description
@	TRANCODE	varchar(255)	
@	STATUS	varchar(255)	
@	CONDITION	varchar(255)	

Uses

	Name
	IPCLOGTRANS_GETTRANSSYNCREFUND
	IPCSYSVAR

Script

```

CREATE procedure [dbo].[IPCLOGTRANS_GETTRANSSYNCREFUND]
    @TRANCODE          VARCHAR (255),
    @STATUS            VARCHAR (255),
    @CONDITION         VARCHAR (255)
AS
BEGIN
    DECLARE @SCRIPT NVARCHAR(4000) = 'SELECT IPCTRANSID AS
PARENTREF, DESTERRORCODE, CCYID, STATUS, ONLINE, CHAR01, CHAR02, CHAR03, CHAR04, CHAR05, CHAR06, CHAR07, CHAR08, CHAR09, CHAR10, CHAR11, CHAR
12, CHAR13, CHAR14, CHAR15, CHAR16, CHAR17, CHAR18, CHAR19, CHAR20, CHAR21, CHAR22, CHAR23, CHAR24, CHAR25, CHAR26, CHAR27, CHAR28, CHAR29, CHA
R30, CHAR31, CHAR32, CHAR33, CHAR34, CHAR35, CHAR36, CHAR37, CHAR38, CHAR39, CHAR40, CHAR41, CHAR42, CHAR43, CHAR44, CHAR45, CHAR46, CHAR47, CH
AR48, CHAR49, CHAR50, CHAR51, CHAR52, CHAR53, CHAR54, CHAR55, CHAR56, CHAR57, CHAR58, CHAR59, CHAR60, CHAR61, CHAR62, CHAR63, CHAR64, CHAR65, C
HAR66, CHAR67, CHAR68, CHAR69, CHAR70, NUM01, NUM02, NUM03, NUM04, NUM05, NUM06, NUM07, NUM08, NUM09, NUM10, NUM11, NUM12, NUM13, NUM14, NUM15, N
UM16, NUM17, NUM18, NUM19, NUM20, NUM21, NUM22, NUM23, NUM24, NUM25, NUM26, NUM27, NUM28, NUM29, NUM30, NUM31, NUM32, NUM33, NUM34, NUM35, NUM36,
NUM37, NUM38, NUM39, NUM40, NUM41, NUM42, NUM43, NUM44, NUM45, NUM46, NUM47, NUM48, NUM49, NUM50, NUM51, NUM52, NUM53, NUM54, NUM55, NUM56, NUM57
, NUM58, NUM59, NUM60, NUM61, NUM62, NUM63, NUM64, NUM65, NUM66, NUM67, NUM68, NUM69, NUM70 FROM IPCLOGTRANS WHERE '
    DECLARE @TRY VARCHAR(20) = (SELECT VARVALUE FROM dbo.IPCSYSVAR WHERE VARNAME='TRETRY09')
    IF (@TRANCODE='')
        BEGIN
            SET @SCRIPT = @SCRIPT + ' STATUS IN ' + @STATUS + ' AND NUM69 < ' + @TRY + ' AND NUM70 = 1 AND CHAR57
IN '+@CONDITION
        END
    ELSE
        BEGIN
            SET @SCRIPT = @SCRIPT + ' STATUS IN ' + @STATUS + ' AND IPCTRANCODE IN ' + @TRANCODE + ' AND NUM69 < '
+ @TRY + ' AND NUM70 = 1 AND CHAR57 IN '+@CONDITION
        END
    EXEC SP_EXECUTE SQL @SCRIPT
END

```

5.2.38. Procedure: IPCLOGTRANS_INSERT

Input/Output

	Name	Data type	Description
@	IPCTRANSDATE	datetime	
@	IPCWORKDATE	varchar(10)	
@	IPCTRANSID	numeric(18, 0)	
@	IPCTRANCODE	varchar(20)	
@	TRANREF	varchar(50)	
@	SENDERID	varchar(20)	
@	RECEIVERID	varchar(20)	
@	SENDERCHANNEL	varchar(20)	
@	RECEIVERCHANNEL	varchar(20)	
@	SOURCEID	varchar(20)	
@	SOURCETRANREF	varchar(50)	
@	USERID	varchar(20)	
@	TRANDESC	nvarchar(400)	
@	STATUS	varchar(1)	
@	APPRSTS	int	
@	OFFLSTS	varchar(1)	
@	DELETED	varchar(1)	
@	ONLINE	bit	
@	ISBATCH	varchar(1)	
@	BATCHREF	varchar(50)	
@	DESTID	varchar(50)	
@	CCYID	varchar(10)	
@	CHAR01	nvarchar(1000)	
@	CHAR02	nvarchar(1000)	
@	CHAR03	nvarchar(1000)	
@	CHAR04	nvarchar(1000)	
@	CHAR05	nvarchar(1000)	
@	CHAR06	nvarchar(1000)	
@	CHAR07	nvarchar(1000)	
@	CHAR08	nvarchar(1000)	
@	CHAR09	nvarchar(1000)	
@	CHAR10	nvarchar(1000)	
@	CHAR11	nvarchar(1000)	
@	CHAR12	nvarchar(1000)	
@	CHAR13	nvarchar(1000)	
@	CHAR14	nvarchar(1000)	
@	CHAR15	nvarchar(1000)	

	Name	Data type	Description
@	CHAR16	nvarchar(1000)	
@	CHAR17	nvarchar(1000)	
@	CHAR18	nvarchar(1000)	
@	CHAR19	nvarchar(1000)	
@	CHAR20	nvarchar(1000)	
@	CHAR21	nvarchar(1000)	
@	CHAR22	nvarchar(1000)	
@	CHAR23	nvarchar(1000)	
@	CHAR24	nvarchar(1000)	
@	CHAR25	nvarchar(1000)	
@	CHAR26	nvarchar(1000)	
@	CHAR27	nvarchar(1000)	
@	CHAR28	nvarchar(1000)	
@	CHAR29	nvarchar(1000)	
@	CHAR30	nvarchar(1000)	
@	CHAR31	nvarchar(1000)	
@	CHAR32	nvarchar(1000)	
@	CHAR33	nvarchar(1000)	
@	CHAR34	nvarchar(1000)	
@	CHAR35	nvarchar(1000)	
@	CHAR36	nvarchar(1000)	
@	CHAR37	nvarchar(1000)	
@	CHAR38	nvarchar(1000)	
@	CHAR39	nvarchar(1000)	
@	CHAR40	nvarchar(1000)	
@	CHAR41	nvarchar(1000)	
@	CHAR42	nvarchar(1000)	
@	CHAR43	nvarchar(1000)	
@	CHAR44	nvarchar(1000)	
@	CHAR45	nvarchar(1000)	
@	CHAR46	nvarchar(1000)	
@	CHAR47	nvarchar(1000)	
@	CHAR48	nvarchar(1000)	
@	CHAR49	nvarchar(1000)	
@	CHAR50	nvarchar(1000)	
@	NUM01	numeric(20, 4)	
@	NUM02	numeric(20, 4)	
@	NUM03	numeric(20, 4)	
@	NUM04	numeric(20, 4)	
@	NUM05	numeric(20, 4)	

	Name	Data type	Description
@NUM06	NUM06	numeric(20, 4)	
@NUM07	NUM07	numeric(20, 4)	
@NUM08	NUM08	numeric(20, 4)	
@NUM09	NUM09	numeric(20, 4)	
@NUM10	NUM10	numeric(20, 4)	
@NUM11	NUM11	numeric(20, 4)	
@NUM12	NUM12	numeric(20, 4)	
@NUM13	NUM13	numeric(20, 4)	
@NUM14	NUM14	numeric(20, 4)	
@NUM15	NUM15	numeric(20, 4)	
@NUM16	NUM16	numeric(20, 4)	
@NUM17	NUM17	numeric(20, 4)	
@NUM18	NUM18	numeric(20, 4)	
@NUM19	NUM19	numeric(20, 4)	
@NUM20	NUM20	numeric(20, 4)	

Uses

	Name
IPCLOGTRANS_INSERT	
IPCLOGTRANS	

Script

```

CREATE procedure [dbo].[IPCLOGTRANS_INSERT]
    @IPCTRANSDATE DATETIME,
    @IPCWORKDATE VARCHAR(10),
    @IPCTRANSID NUMERIC(18, 0),
    @IPCTRANCODE VARCHAR(20),
    @TRANREF VARCHAR(50),
    @SENDERID VARCHAR(20) = '',
    @RECEIVERID VARCHAR(20) = '',
    @SENDERCHANNEL VARCHAR(20) = '',
    @RECEIVERCHANNEL VARCHAR(20) = '',
    @SOURCEID VARCHAR(20),
    @SOURCETRANREF VARCHAR(50),
    @USERID VARCHAR(20),
    @TRANDESC NVARCHAR(400),
    @STATUS VARCHAR(1),
    @APPRSTS INT,
    @OFFLSTS VARCHAR(1),
    @DELETED VARCHAR(1),
    @ONLINE BIT,
    @ISBATCH VARCHAR(1) = NULL,
    @BATCHREF VARCHAR(50) = NULL,
    @DESTID VARCHAR(50) = NULL,
    @CCYID VARCHAR(10) = NULL,
    @CHAR01 NVARCHAR(1000),
    @CHAR02 NVARCHAR(1000),
    @CHAR03 NVARCHAR(1000),
    @CHAR04 NVARCHAR(1000),
    @CHAR05 NVARCHAR(1000),
    @CHAR06 NVARCHAR(1000),
    @CHAR07 NVARCHAR(1000),
    @CHAR08 NVARCHAR(1000),
    @CHAR09 NVARCHAR(1000),
    @CHAR10 NVARCHAR(1000),
    @CHAR11 NVARCHAR(1000),
    @CHAR12 NVARCHAR(1000),
    @CHAR13 NVARCHAR(1000),
    @CHAR14 NVARCHAR(1000),
    @CHAR15 NVARCHAR(1000),
    @CHAR16 NVARCHAR(1000),
    @CHAR17 NVARCHAR(1000),
    @CHAR18 NVARCHAR(1000),
    @CHAR19 NVARCHAR(1000),
    @CHAR20 NVARCHAR(1000),

```

```

@CHAR21 NVARCHAR(1000),
@CHAR22 NVARCHAR(1000),
@CHAR23 NVARCHAR(1000),
@CHAR24 NVARCHAR(1000),
@CHAR25 NVARCHAR(1000),
@CHAR26 NVARCHAR(1000),
@CHAR27 NVARCHAR(1000),
@CHAR28 NVARCHAR(1000),
@CHAR29 NVARCHAR(1000),
@CHAR30 NVARCHAR(1000),
@CHAR31 NVARCHAR(1000),
@CHAR32 NVARCHAR(1000),
@CHAR33 NVARCHAR(1000),
@CHAR34 NVARCHAR(1000),
@CHAR35 NVARCHAR(1000),
@CHAR36 NVARCHAR(1000),
@CHAR37 NVARCHAR(1000),
@CHAR38 NVARCHAR(1000),
@CHAR39 NVARCHAR(1000),
@CHAR40 NVARCHAR(1000),
@CHAR41 NVARCHAR(1000),
@CHAR42 NVARCHAR(1000),
@CHAR43 NVARCHAR(1000),
@CHAR44 NVARCHAR(1000),
@CHAR45 NVARCHAR(1000),
@CHAR46 NVARCHAR(1000),
@CHAR47 NVARCHAR(1000),
@CHAR48 NVARCHAR(1000),
@CHAR49 NVARCHAR(1000),
@CHAR50 NVARCHAR(1000),

@NUM01 NUMERIC(20,4),
@NUM02 NUMERIC(20,4),
@NUM03 NUMERIC(20,4),
@NUM04 NUMERIC(20,4),
@NUM05 NUMERIC(20,4),
@NUM06 NUMERIC(20,4),
@NUM07 NUMERIC(20,4),
@NUM08 NUMERIC(20,4),
@NUM09 NUMERIC(20,4),
@NUM10 NUMERIC(20,4),
@NUM11 NUMERIC(20,4),
@NUM12 NUMERIC(20,4),
@NUM13 NUMERIC(20,4),
@NUM14 NUMERIC(20,4),
@NUM15 NUMERIC(20,4),
@NUM16 NUMERIC(20,4),
@NUM17 NUMERIC(20,4),
@NUM18 NUMERIC(20,4),
@NUM19 NUMERIC(20,4),
@NUM20 NUMERIC(20,4)

AS
BEGIN
    SET DATEFORMAT DMY
    INSERT INTO IPCLOGTRANS(IPCTRANSDATE, IPCWORKDATE, IPCTRANSID, IPCTRANCODE, TRANREF, SOURCEID, SOURCETRANREF,
                           USERID, TRANDESC, STATUS, APPRSTS, OFFLSTS, DELETED, DESTID, ONLINE, SENDERBANK,
                           RECEIVERBANK, SENDERCHANNEL, RECEIVERCHANNEL,
                           CHAR01, CHAR02, CHAR03, CHAR04, CHAR05, CHAR06, CHAR07, CHAR08, CHAR09, CHAR10,
                           CHAR11, CHAR12, CHAR13, CHAR14, CHAR15, CHAR16, CHAR17, CHAR18, CHAR19, CHAR20,
                           CHAR21, CHAR22, CHAR23, CHAR24, CHAR25, CHAR26, CHAR27, CHAR28, CHAR29, CHAR30,
                           CHAR31, CHAR32, CHAR33, CHAR34, CHAR35, CHAR36, CHAR37, CHAR38, CHAR39, CHAR40,
                           CHAR41, CHAR42, CHAR43, CHAR44, CHAR45, CHAR46, CHAR47, CHAR48, CHAR49, CHAR50,
                           NUM01, NUM02, NUM03, NUM04, NUM05, NUM06, NUM07, NUM08, NUM09, NUM10,
                           NUM11, NUM12, NUM13, NUM14, NUM15, NUM16, NUM17, NUM18, NUM19, NUM20,
                           CCYID, ISBATCH, BATCHREF)
        VALUES (@IPCTRANSDATE, @IPCWORKDATE, @IPCTRANSID, @IPCTRANCODE, @TRANREF, @SOURCEID, ISNULL(@SOURCETRANREF, ''),
                @USERID, ISNULL(@TRANDESC, ''), @STATUS, @APPRSTS, @OFFLSTS, @DELETED, ISNULL(@DESTID, ''),
                @ONLINE, @SENDERID, @RECEIVERID, @SENDERCHANNEL, @RECEIVERCHANNEL,
                @CHAR01, @CHAR02, @CHAR03, @CHAR04, @CHAR05, @CHAR06, @CHAR07, @CHAR08, @CHAR09, @CHAR10,
                @CHAR11, @CHAR12, @CHAR13, @CHAR14, @CHAR15, @CHAR16, @CHAR17, @CHAR18, @CHAR19, @CHAR20,
                @CHAR21, @CHAR22, @CHAR23, @CHAR24, @CHAR25, @CHAR26, @CHAR27, @CHAR28, @CHAR29, @CHAR30,
                @CHAR31, @CHAR32, @CHAR33, @CHAR34, @CHAR35, @CHAR36, @CHAR37, @CHAR38, @CHAR39, @CHAR40,
                @CHAR41, @CHAR42, @CHAR43, @CHAR44, @CHAR45, @CHAR46, @CHAR47, @CHAR48, @CHAR49, @CHAR50,
                @NUM01, @NUM02, @NUM03, @NUM04, @NUM05, @NUM06, @NUM07, @NUM08, @NUM09, @NUM10,
                @NUM11, @NUM12, @NUM13, @NUM14, @NUM15, @NUM16, @NUM17, @NUM18, @NUM19, @NUM20,
                @CCYID, @ISBATCH, @BATCHREF)
END

```

5.2.39. Procedure: IPCLOGTRANS_INSERT_EVENT

Input/Output

	Name	Data type	Description
@	IPCTRANSDATE	datetime	
@	IPCWORKDATE	varchar(10)	
@	IPCTRANSID	numeric(18, 0)	
@	IPCTRANCODE	varchar(20)	
@	SOURCEID	varchar(20)	
@	SOURCETRANREF	varchar(50)	
@	USERID	varchar(20)	
@	TRANDESC	nvarchar(400)	
@	STATUS	varchar(10)	
@	APPRSTS	int	
@	OFFLSTS	varchar(1)	
@	DELETED	varchar(1)	
@	ONLINE	bit	
@	ISBATCH	varchar(1)	
@	BATCHREF	varchar(50)	
@	DESTID	varchar(50)	
@	CCYID	varchar(10)	
@	PARENT	varchar(18)	
@	CHAR01	nvarchar(1000)	
@	CHAR02	nvarchar(1000)	
@	CHAR03	nvarchar(1000)	
@	CHAR04	nvarchar(1000)	
@	CHAR05	nvarchar(1000)	
@	CHAR06	nvarchar(1000)	
@	CHAR07	nvarchar(1000)	
@	CHAR08	nvarchar(1000)	
@	CHAR09	nvarchar(1000)	
@	CHAR10	nvarchar(1000)	
@	CHAR11	nvarchar(1000)	
@	CHAR12	nvarchar(1000)	
@	CHAR13	nvarchar(1000)	
@	CHAR14	nvarchar(1000)	
@	CHAR15	nvarchar(1000)	
@	CHAR16	nvarchar(1000)	
@	CHAR17	nvarchar(1000)	
@	CHAR18	nvarchar(1000)	
@	CHAR19	nvarchar(1000)	

	Name	Data type	Description
>@	CHAR20	nvarchar(1000)	
>@	CHAR21	nvarchar(1000)	
>@	CHAR22	nvarchar(1000)	
>@	CHAR23	nvarchar(1000)	
>@	CHAR24	nvarchar(1000)	
>@	CHAR25	nvarchar(1000)	
>@	CHAR26	nvarchar(1000)	
>@	CHAR27	nvarchar(1000)	
>@	CHAR28	nvarchar(1000)	
>@	CHAR29	nvarchar(1000)	
>@	CHAR30	nvarchar(1000)	
>@	CHAR31	nvarchar(1000)	
>@	CHAR32	nvarchar(1000)	
>@	CHAR33	nvarchar(1000)	
>@	CHAR34	nvarchar(1000)	
>@	CHAR35	nvarchar(1000)	
>@	CHAR36	nvarchar(1000)	
>@	CHAR37	nvarchar(1000)	
>@	CHAR38	nvarchar(1000)	
>@	CHAR39	nvarchar(1000)	
>@	CHAR40	nvarchar(1000)	
>@	CHAR41	nvarchar(1000)	
>@	CHAR42	nvarchar(1000)	
>@	CHAR43	nvarchar(1000)	
>@	CHAR44	nvarchar(1000)	
>@	CHAR45	nvarchar(1000)	
>@	CHAR46	nvarchar(1000)	
>@	CHAR47	nvarchar(1000)	
>@	CHAR48	nvarchar(1000)	
>@	CHAR49	nvarchar(1000)	
>@	CHAR50	nvarchar(1000)	
>@	CHAR51	nvarchar(1000)	
>@	CHAR52	nvarchar(1000)	
>@	CHAR53	nvarchar(1000)	
>@	CHAR54	nvarchar(1000)	
>@	CHAR55	nvarchar(1000)	
>@	CHAR56	nvarchar(1000)	
>@	CHAR57	nvarchar(1000)	
>@	CHAR58	nvarchar(1000)	
>@	CHAR59	nvarchar(1000)	

	Name	Data type	Description
@	CHAR60	nvarchar(1000)	
@	CHAR61	nvarchar(1000)	
@	CHAR62	nvarchar(1000)	
@	CHAR63	nvarchar(1000)	
@	CHAR64	nvarchar(1000)	
@	CHAR65	nvarchar(1000)	
@	CHAR66	nvarchar(1000)	
@	CHAR67	nvarchar(1000)	
@	CHAR68	nvarchar(1000)	
@	CHAR69	nvarchar(1000)	
@	CHAR70	nvarchar(1000)	
@	NUM01	numeric(20, 4)	
@	NUM02	numeric(20, 4)	
@	NUM03	numeric(20, 4)	
@	NUM04	numeric(20, 4)	
@	NUM05	numeric(20, 4)	
@	NUM06	numeric(20, 4)	
@	NUM07	numeric(20, 4)	
@	NUM08	numeric(20, 4)	
@	NUM09	numeric(20, 4)	
@	NUM10	numeric(20, 4)	
@	NUM11	numeric(20, 4)	
@	NUM12	numeric(20, 4)	
@	NUM13	numeric(20, 4)	
@	NUM14	numeric(20, 4)	
@	NUM15	numeric(20, 4)	
@	NUM16	numeric(20, 4)	
@	NUM17	numeric(20, 4)	
@	NUM18	numeric(20, 4)	
@	NUM19	numeric(20, 4)	
@	NUM20	numeric(20, 4)	
@	NUM21	numeric(20, 4)	
@	NUM22	numeric(20, 4)	
@	NUM23	numeric(20, 4)	
@	NUM24	numeric(20, 4)	
@	NUM25	numeric(20, 4)	
@	NUM26	numeric(20, 4)	
@	NUM27	numeric(20, 4)	
@	NUM28	numeric(20, 4)	
@	NUM29	numeric(20, 4)	

	Name	Data type	Description
@	NUM30	numeric(20, 4)	
@	NUM31	numeric(20, 4)	
@	NUM32	numeric(20, 4)	
@	NUM33	numeric(20, 4)	
@	NUM34	numeric(20, 4)	
@	NUM35	numeric(20, 4)	
@	NUM36	numeric(20, 4)	
@	NUM37	numeric(20, 4)	
@	NUM38	numeric(20, 4)	
@	NUM39	numeric(20, 4)	
@	NUM40	numeric(20, 4)	
@	NUM41	numeric(20, 4)	
@	NUM42	numeric(20, 4)	
@	NUM43	numeric(20, 4)	
@	NUM44	numeric(20, 4)	
@	NUM45	numeric(20, 4)	
@	NUM46	numeric(20, 4)	
@	NUM47	numeric(20, 4)	
@	NUM48	numeric(20, 4)	
@	NUM49	numeric(20, 4)	
@	NUM50	numeric(20, 4)	
@	NUM51	numeric(20, 4)	
@	NUM52	numeric(20, 4)	
@	NUM53	numeric(20, 4)	
@	NUM54	numeric(20, 4)	
@	NUM55	numeric(20, 4)	
@	NUM56	numeric(20, 4)	
@	NUM57	numeric(20, 4)	
@	NUM58	numeric(20, 4)	
@	NUM59	numeric(20, 4)	
@	NUM60	numeric(20, 4)	
@	NUM61	numeric(20, 4)	
@	NUM62	numeric(20, 4)	
@	NUM63	numeric(20, 4)	
@	NUM64	numeric(20, 4)	
@	NUM65	numeric(20, 4)	
@	NUM66	numeric(20, 4)	
@	NUM67	numeric(20, 4)	
@	NUM68	numeric(20, 4)	
@	NUM69	numeric(20, 4)	

Name	Data type	Description
@NUM70	numeric(20, 4)	

Uses

Name
IPCLOGTRANS_INSERT_EVENT
IPCLOGTRANSEVENT

Script

```
-- =====
-- Author: Admin
-- Create date: 14/02/2009
-- Description: Log Transaction Info
-- =====
CREATE procedure [dbo].[IPCLOGTRANS_INSERT_EVENT]
    @IPCTRANSDATE DATETIME,
    @IPCTWORKDATE VARCHAR(10),
    @IPCTRANSID NUMERIC(18, 0),
    @IPCTRANCODE VARCHAR(20),
    @SOURCEID VARCHAR(20),
    @SOURCETRANREF VARCHAR(50),
    @USERID VARCHAR(20),
    @TRANDESC NVARCHAR(400),
    @STATUS VARCHAR(10),
    @APPRSTS INT,
    @OFFLSTS VARCHAR(1),
    @DELETED VARCHAR(1),
    @ONLINE BIT,
    @ISBATCH VARCHAR(1) = NULL,
    @BATCHREF VARCHAR(50) = NULL,
    @DESTID VARCHAR(50) = NULL,
    @CCYID VARCHAR(10) = NULL,
    @PARENT VARCHAR(18) = NULL,
    @CHAR01 NVARCHAR(1000),
    @CHAR02 NVARCHAR(1000),
    @CHAR03 NVARCHAR(1000),
    @CHAR04 NVARCHAR(1000),
    @CHAR05 NVARCHAR(1000),
    @CHAR06 NVARCHAR(1000),
    @CHAR07 NVARCHAR(1000),
    @CHAR08 NVARCHAR(1000),
    @CHAR09 NVARCHAR(1000),
    @CHAR10 NVARCHAR(1000),
    @CHAR11 NVARCHAR(1000),
    @CHAR12 NVARCHAR(1000),
    @CHAR13 NVARCHAR(1000),
    @CHAR14 NVARCHAR(1000),
    @CHAR15 NVARCHAR(1000),
    @CHAR16 NVARCHAR(1000),
    @CHAR17 NVARCHAR(1000),
    @CHAR18 NVARCHAR(1000),
    @CHAR19 NVARCHAR(1000),
    @CHAR20 NVARCHAR(1000),
    @CHAR21 NVARCHAR(1000),
    @CHAR22 NVARCHAR(1000),
    @CHAR23 NVARCHAR(1000),
    @CHAR24 NVARCHAR(1000),
    @CHAR25 NVARCHAR(1000),
    @CHAR26 NVARCHAR(1000),
    @CHAR27 NVARCHAR(1000),
    @CHAR28 NVARCHAR(1000),
    @CHAR29 NVARCHAR(1000),
    @CHAR30 NVARCHAR(1000),
    @CHAR31 NVARCHAR(1000),
    @CHAR32 NVARCHAR(1000),
    @CHAR33 NVARCHAR(1000),
    @CHAR34 NVARCHAR(1000),
    @CHAR35 NVARCHAR(1000),
    @CHAR36 NVARCHAR(1000),
    @CHAR37 NVARCHAR(1000),
    @CHAR38 NVARCHAR(1000),
    @CHAR39 NVARCHAR(1000),
    @CHAR40 NVARCHAR(1000),
    @CHAR41 NVARCHAR(1000),
    @CHAR42 NVARCHAR(1000),
    @CHAR43 NVARCHAR(1000),
    @CHAR44 NVARCHAR(1000),
    @CHAR45 NVARCHAR(1000),
    @CHAR46 NVARCHAR(1000),
    @CHAR47 NVARCHAR(1000),
    @CHAR48 NVARCHAR(1000),
    @CHAR49 NVARCHAR(1000),
    @CHAR50 NVARCHAR(1000),
```

```
    @CHAR51          NVARCHAR(1000),  
    @CHAR52          NVARCHAR(1000),  
    @CHAR53          NVARCHAR(1000),  
    @CHAR54          NVARCHAR(1000),  
    @CHAR55          NVARCHAR(1000),  
    @CHAR56          NVARCHAR(1000),  
    @CHAR57          NVARCHAR(1000),  
    @CHAR58          NVARCHAR(1000),  
    @CHAR59          NVARCHAR(1000),  
    @CHAR60          NVARCHAR(1000),  
    @CHAR61          NVARCHAR(1000),  
    @CHAR62          NVARCHAR(1000),  
    @CHAR63          NVARCHAR(1000),  
    @CHAR64          NVARCHAR(1000),  
    @CHAR65          NVARCHAR(1000),  
    @CHAR66          NVARCHAR(1000),  
    @CHAR67          NVARCHAR(1000),  
    @CHAR68          NVARCHAR(1000),  
    @CHAR69          NVARCHAR(1000),  
    @CHAR70          NVARCHAR(1000),  
  
    @NUM01          NUMERIC(20,4),  
    @NUM02          NUMERIC(20,4),  
    @NUM03          NUMERIC(20,4),  
    @NUM04          NUMERIC(20,4),  
    @NUM05          NUMERIC(20,4),  
    @NUM06          NUMERIC(20,4),  
    @NUM07          NUMERIC(20,4),  
    @NUM08          NUMERIC(20,4),  
    @NUM09          NUMERIC(20,4),  
    @NUM10          NUMERIC(20,4),  
    @NUM11          NUMERIC(20,4),  
    @NUM12          NUMERIC(20,4),  
    @NUM13          NUMERIC(20,4),  
    @NUM14          NUMERIC(20,4),  
    @NUM15          NUMERIC(20,4),  
    @NUM16          NUMERIC(20,4),  
    @NUM17          NUMERIC(20,4),  
    @NUM18          NUMERIC(20,4),  
    @NUM19          NUMERIC(20,4),  
    @NUM20          NUMERIC(20,4),  
    @NUM21          NUMERIC(20,4),  
    @NUM22          NUMERIC(20,4),  
    @NUM23          NUMERIC(20,4),  
    @NUM24          NUMERIC(20,4),  
    @NUM25          NUMERIC(20,4),  
    @NUM26          NUMERIC(20,4),  
    @NUM27          NUMERIC(20,4),  
    @NUM28          NUMERIC(20,4),  
    @NUM29          NUMERIC(20,4),  
    @NUM30          NUMERIC(20,4),  
    @NUM31          NUMERIC(20,4),  
    @NUM32          NUMERIC(20,4),  
    @NUM33          NUMERIC(20,4),  
    @NUM34          NUMERIC(20,4),  
    @NUM35          NUMERIC(20,4),  
    @NUM36          NUMERIC(20,4),  
    @NUM37          NUMERIC(20,4),  
    @NUM38          NUMERIC(20,4),  
    @NUM39          NUMERIC(20,4),  
    @NUM40          NUMERIC(20,4),  
    @NUM41          NUMERIC(20,4),  
    @NUM42          NUMERIC(20,4),  
    @NUM43          NUMERIC(20,4),  
    @NUM44          NUMERIC(20,4),  
    @NUM45          NUMERIC(20,4),  
    @NUM46          NUMERIC(20,4),  
    @NUM47          NUMERIC(20,4),  
    @NUM48          NUMERIC(20,4),  
    @NUM49          NUMERIC(20,4),  
    @NUM50          NUMERIC(20,4),  
    @NUM51          NUMERIC(20,4),  
  
    @NUM52          NUMERIC(20,4),  
    @NUM53          NUMERIC(20,4),  
    @NUM54          NUMERIC(20,4),  
    @NUM55          NUMERIC(20,4),  
    @NUM56          NUMERIC(20,4),  
    @NUM57          NUMERIC(20,4),  
    @NUM58          NUMERIC(20,4),  
    @NUM59          NUMERIC(20,4),  
    @NUM60          NUMERIC(20,4),  
    @NUM61          NUMERIC(20,4),  
    @NUM62          NUMERIC(20,4),  
    @NUM63          NUMERIC(20,4),  
    @NUM64          NUMERIC(20,4),  
    @NUM65          NUMERIC(20,4),  
    @NUM66          NUMERIC(20,4),  
    @NUM67          NUMERIC(20,4),  
    @NUM68          NUMERIC(20,4),  
    @NUM69          NUMERIC(20,4),  
    @NUM70          NUMERIC(20,4)
```

```

AS
BEGIN
    SET DATEFORMAT DMY
    INSERT INTO IPCLOGTRANSEVENT(IPCTRANSDATE,IPCWORKDATE,IPCTRANSID,IPCTRANCODE,SOURCEID,SOURCETRANREF,
        USERID,TRANDESC,STATUS,APPRSTS,OFLSTS,DELETED,DESTID,ONLINE,PARENTREF,
        CHAR01,CHAR02,CHAR03,CHAR04,CHAR05,CHAR06,CHAR07,CHAR08,CHAR09,CHAR10,
        CHAR11,CHAR12,CHAR13,CHAR14,CHAR15,CHAR16,CHAR17,CHAR18,CHAR19,CHAR20,
        CHAR21,CHAR22,CHAR23,CHAR24,
        CHAR25,CHAR26,CHAR27,CHAR28,CHAR29,CHAR30,CHAR31,CHAR32,CHAR33,CHAR34,CHAR35,CHAR36,CHAR37,CHAR38,CHAR39,CHAR40,CHAR41,CHAR42
        ,CHAR43,CHAR44,CHAR45,CHAR46,CHAR47,CHAR48,CHAR49,CHAR50,CHAR51,CHAR52,CHAR53,CHAR54,CHAR55,CHAR56,CHAR57,CHAR58,CHAR59,CHAR6
        0,CHAR61,CHAR62,CHAR63,CHAR64,CHAR65,CHAR66,CHAR67,CHAR68,CHAR69,CHAR70,
        NUM01,NUM02,NUM03,NUM04,NUM05,NUM06,NUM07,NUM08,NUM09,NUM10,
        NUM11,NUM12,NUM13,NUM14,NUM15,NUM16,NUM17,NUM18,NUM19,NUM20,
        NUM21,NUM22,NUM23,NUM24,
        CCYID,ISBATCH,BATCHREF)
    VALUES (@IPCTRANSDATE,@IPCWORKDATE,@IPCTRANSID,@IPCTRANCODE,@SOURCEID,ISNULL(@SOURCETRANREF,''),
        @USERID,ISNULL(@TRANDESC,''),@STATUS,@APPRSTS,@OFLSTS,@DELETED,ISNULL(@DESTID,''),
        @ONLINE,@PARENT,
        @CHAR01,@CHAR02,@CHAR03,@CHAR04,@CHAR05,@CHAR06,@CHAR07,@CHAR08,@CHAR09,@CHAR10,
        @CHAR11,@CHAR12,@CHAR13,@CHAR14,@CHAR15,@CHAR16,@CHAR17,@CHAR18,@CHAR19,@CHAR20,
        @CHAR21,@CHAR22,@CHAR23,@CHAR24,
        @CHAR25,@CHAR26,@CHAR27,@CHAR28,@CHAR29,@CHAR30,@CHAR31,@CHAR32,@CHAR33,@CHAR34,@CHAR35,@CHAR36,@CHAR37,@CHAR38,@CHAR39,@CHAR
        40,@CHAR41,@CHAR42,@CHAR43,@CHAR44,@CHAR45,@CHAR46,@CHAR47,@CHAR48,@CHAR49,@CHAR50,@CHAR51,@CHAR52,@CHAR53,@CHAR54,@CHAR55,@C
        HAR56,@CHAR57,@CHAR58,@CHAR59,@CHAR60,@CHAR61,@CHAR62,@CHAR63,@CHAR64,@CHAR65,@CHAR66,@CHAR67,@CHAR68,@CHAR69,@CHAR70,
        @NUM01,@NUM02,@NUM03,@NUM04,@NUM05,@NUM06,@NUM07,@NUM08,@NUM09,@NUM10,
        @NUM11,@NUM12,@NUM13,@NUM14,@NUM15,@NUM16,@NUM17,@NUM18,@NUM19,@NUM20,
        @NUM21,@NUM22,@NUM23,@NUM24,
        @NUM25,@NUM26,@NUM27,@NUM28,@NUM29,@NUM30,@NUM31,@NUM32,@NUM33,@NUM34,@NUM35,@NUM36,@NUM37,@NUM38,@NUM39,@NUM40,@NUM41,@NUM42
        ,@NUM43,@NUM44,@NUM45,@NUM46,@NUM47,@NUM48,@NUM49,@NUM50,@NUM51,@NUM52,@NUM53,@NUM54,@NUM55,@NUM56,@NUM57,@NUM58,@NUM59,@NUM6
        0,@NUM61,@NUM62,@NUM63,@NUM64,@NUM65,@NUM66,@NUM67,@NUM68,@NUM69,@NUM70,
        @CCYID,@ISBATCH,@BATCHREF)
END

```

RTIA

5.2.40. Procedure: IPCLOGTRANS_OFFLINE_SELECT

Uses

Name
IPCLOGTRANS_OFFLINE_SELECT
IPCLOGTRANS

Script

```
-- =====
-- Author:      Admin
-- Create date: 14/02/2009
-- Description: Get Offline Transaction
-- =====
CREATE procedure [dbo].[IPCLOGTRANS_OFFLINE_SELECT]
AS
BEGIN
    SELECT A.IPCTRANSID, A.IPCTRANCODE, A.TRANDESC, A.SOURCEID, A.SOURCETRANREF, A.DESTID, A.DESTTRANREF
    FROM IPCLOGTRANS A (NOLOCK)
    WHERE A.ONLINE = 0 AND A.DELETED = 0 AND A.STATUS = '1'
          AND A.OFFLSTS IN ('', '0', '2')
END
```

TRIAL

5.2.41. Procedure: IPCLOGTRANS_OFFLINE_UPDATE

Input/Output

	Name	Data type	Description
@	IPCWORKDATE	varchar(10)	
@	IPCTRANSID	numeric(18, 0)	
@	OFFLSTS	varchar(1)	
@	DESTTRANREF	varchar(20)	
@	ERRORCODE	varchar(10)	
@	ERRORDESC	nvarchar(1000)	

Uses

Name
IPCLOGTRANS_OFFLINE_UPDATE
IPCLOGTRANS

Script

```
-- =====
-- Author:          Admin
-- Create date:    14/02/2009
-- Description:    Update Offline Transaction Info
-- =====
CREATE procedure [dbo].[IPCLOGTRANS_OFFLINE_UPDATE]
    @IPCWORKDATE      VARCHAR(10),
    @IPCTRANSID       NUMERIC(18, 0),
    @OFFLSTS          VARCHAR(1),
    @DESTTRANREF      VARCHAR(20),
    @ERRORCODE         VARCHAR(10),
    @ERRORDESC        NVARCHAR(1000)
AS
BEGIN
    SET DATEFORMAT DMY
    UPDATE IPCLOGTRANS
    SET OFFLSTS = @OFFLSTS, DESTTRANREF = @DESTTRANREF,
        ERRORCODE = @ERRORCODE, ERRORDESC = @ERRORDESC
    WHERE IPCWORKDATE = @IPCWORKDATE AND IPCTRANSID = @IPCTRANSID
END
```

5.2.42. Procedure: IPCLOGTRANS_REVERSAL

Input/Output

Name		Data type	Description
@IPCWORKDATE	IPCWORKDATE	varchar(10)	
@IPCTRANSID	IPCTRANSID	numeric(18, 0)	
@IPCTRANSIDREVERAL	IPCTRANSIDREVERAL	numeric(18, 0)	

Uses

Name
IPCLOGTRANS_REVERSAL
IPCLOGTRANS

Script

```
-- =====
-- Author: Admin
-- Create date: 14/02/2009
-- Description: Reversal Transaction
-- =====
CREATE procedure [dbo].[IPCLOGTRANS_REVERSAL]
    @IPCWORKDATE           VARCHAR(10),
    @IPCTRANSID             NUMERIC(18, 0),
    @IPCTRANSIDREVERAL     NUMERIC(18, 0)
AS
BEGIN
    SET DATEFORMAT DMY
    UPDATE IPCLOGTRANS SET DELETED = 1
    WHERE IPCWORKDATE = @IPCWORKDATE
        AND (IPCTRANSID = @IPCTRANSID OR IPCTRANSID = @IPCTRANSIDREVERAL)
END
```

5.2.43. Procedure: IPCLOGTRANS_SYNC09_UPDATE

Input/Output

	Name	Data type	Description
@	IPCTRANSID	numeric(18, 0)	
@	ERRORCODE	varchar(10)	
@	ERRORDESC	nvarchar(1000)	

Uses

Name
IPCLOGTRANS_SYNC09_UPDATE
IPCLOGTRANS

Script

```

CREATE procedure [dbo].[IPCLOGTRANS_SYNC09_UPDATE]
    @IPCTRANSID           NUMERIC(18, 0),
    @ERRORCODE             VARCHAR(10),
    @ERRORDESC              NVARCHAR(1000)
AS
BEGIN
--30.8.2017 minh modify this
--get char49 and top line
--declare @char49 as varchar(1000)
declare @topline as varchar(200)
--select @char49= CHAR49,@topline=SUBSTRING(CHAR49,1,case when CHARINDEX(CHAR(13),CHAR49,1)=0 then len(char49) else
CHARINDEX(CHAR(13),CHAR49,1) end ) from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID
select @topline=SUBSTRING(CHAR49,1,case when CHARINDEX(CHAR(13),CHAR49,1)=0 then len(char49) else
CHARINDEX(CHAR(13),CHAR49,1)-1 end ) from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID

    IF(@ERRORCODE ='0')
        BEGIN
            SET @ERRORDESC='SYNCHED TO O9 CORE'
            IF(@ERRORDESC=@topline)
                UPDATE IPCLOGTRANS SET NUM70=2 WHERE IPCTRANSID = @IPCTRANSID
            ELSE
                UPDATE IPCLOGTRANS SET NUM70=2,CHAR49=case when @topline='' then @ERRORDESC else @ERRORDESC +
+ CHAR(13) + CHAR49 end WHERE IPCTRANSID = @IPCTRANSID
            END
        ELSE
            BEGIN
                IF(@ERRORDESC=@topline)
                    UPDATE IPCLOGTRANS SET NUM69=NUM69 + 1 WHERE IPCTRANSID = @IPCTRANSID
                ELSE
                    UPDATE IPCLOGTRANS SET CHAR49=case when @topline='' then @ERRORDESC else @ERRORDESC +
+ CHAR(13) + CHAR49 end, NUM69=NUM69 + 1 WHERE IPCTRANSID = @IPCTRANSID
            END
    END

```

5.2.44. Procedure: IPCLOGTRANS_TIMEOUT

Uses

Name
IPCLOGTRANS_TIMEOUT
IPC_TIMEOUTTRANS
IPCLOGTRANS

Script

```
CREATE proc [dbo].[IPCLOGTRANS_TIMEOUT]
AS
BEGIN
    UPDATE IPCLOGTRANS SET STATUS = '2', APPRSTS = '10'
    WHERE IPCTRANSID IN
        (SELECT IPCTRANSID FROM IPCLOGTRANS A, IPC_TIMEOUTTRANS B
         WHERE A.IPCTRANCODE = B.IPCTRANCODE
         AND STATUS = '3' AND APPRSTS = '4'
         AND DATEDIFF(minute,A.IPCTRANSDATE,GETDATE()) > B.TIMEOUT)
END
```

TRIAL

5.2.45. Procedure: IPCLOGTRANS_UPDATE

Input/Output

	Name	Data type	Description
@	IPCWORKDATE	varchar(10)	
@	IPCTRANSID	numeric(18, 0)	
@	STATUS	varchar(1)	
@	APPRSTS	varchar(1)	
@	OFFLSTS	varchar(1)	
@	DELETED	varchar(1)	
@	DESTID	varchar(20)	
@	DESTTRANREF	varchar(20)	
@	DESTERRORCODE	nvarchar(MAX)	
@	ERRORCODE	varchar(10)	
@	ERRORDESC	nvarchar(1000)	
@	ONLINE	bit	

Uses

Name
IPCLOGTRANS_UPDATE
IPCLOGTRANS

Script

```
-- =====
-- Author:          Admin
-- Create date: 14/02/2009
-- Description:    Update Log Transaction Info
-- =====
CREATE procedure [dbo].[IPCLOGTRANS_UPDATE]
    @IPCWORKDATE           VARCHAR(10),
    @IPCTRANSID             NUMERIC(18, 0),
    @STATUS                 VARCHAR(1),
    @APPRSTS                VARCHAR(1),
    @OFFLSTS                VARCHAR(1),
    @DELETED                VARCHAR(1),
    @DESTID                 VARCHAR(20),
    @DESTTRANREF            VARCHAR(20),
    @DESTERRORCODE          NVARCHAR(MAX),
    @ERRORCODE               VARCHAR(10),
    @ERRORDESC               NVARCHAR(1000),
    @ONLINE                 BIT
AS
BEGIN
    SET DATEFORMAT DMY
    UPDATE IPCLOGTRANS
    SET STATUSHIS = STATUSHIS + '#' + STATUS, STATUS = @STATUS, APPRSTS = @APPRSTS, OFFLSTS = @OFFLSTS,
        DELETED = @DELETED, DESTID = @DESTID, DESTTRANREF = @DESTTRANREF,
        DESTERRORCODE = @DESTERRORCODE, ERRORCODE = @ERRORCODE,
        ERRORDESC = @ERRORDESC, ONLINE = @ONLINE
    WHERE IPCTRANSID = @IPCTRANSID
END
```

5.2.46. Procedure: IPCLOGTRANS_UPDATE_EVENT

Input/Output

	Name	Data type	Description
@@	IPCWORKDATE	varchar(10)	
@@	IPCTRANSID	numeric(18, 0)	
@@	STATUS	varchar(10)	
@@	APPRSTS	varchar(1)	
@@	OFFLSTS	varchar(1)	
@@	DELETED	varchar(1)	
@@	DESTID	varchar(20)	
@@	DESTTRANREF	varchar(20)	
@@	DESTERRORCODE	nvarchar(MAX)	
@@	ERRORCODE	varchar(10)	
@@	ERRORDESC	nvarchar(1000)	
@@	ONLINE	bit	

Uses

Name
IPCLOGTRANS_UPDATE_EVENT
IPCLOGTRANSEVENT

Script

```
-- =====
-- Author:          Admin
-- Create date: 14/02/2009
-- Description:    Update Log Transaction Info
-- =====
CREATE procedure [dbo].[IPCLOGTRANS_UPDATE_EVENT]
    @IPCWORKDATE           VARCHAR(10),
    @IPCTRANSID            NUMERIC(18, 0),
    @STATUS                 VARCHAR(10),
    @APPRSTS                VARCHAR(1),
    @OFFLSTS                VARCHAR(1),
    @DELETED                VARCHAR(1),
    @DESTID                 VARCHAR(20),
    @DESTTRANREF            VARCHAR(20),
    @DESTERRORCODE          NVARCHAR(MAX),
    @ERRORCODE               VARCHAR(10),
    @ERRORDESC               NVARCHAR(1000),
    @ONLINE                 BIT
AS
BEGIN
    SET DATEFORMAT DMY
    UPDATE IPCLOGTRANSEVENT
    SET STATUSHIS = STATUSHIS + '#' + STATUS, STATUS = @STATUS, APPRSTS = @APPRSTS, OFFLSTS = @OFFLSTS,
        DELETED = @DELETED, DESTID = @DESTID, DESTTRANREF = @DESTTRANREF,
        DESTERRORCODE = @DESTERRORCODE, ERRORCODE = @ERRORCODE,
        ERRORDESC = @ERRORDESC, ONLINE = @ONLINE
    WHERE IPCTRANSID = @IPCTRANSID
END
```

5.2.47. Procedure: IPCLOGTRANS_UPDATE_REVERSE_REFERENCE

Input/Output

Name	Data type	Description
@IPCTRANSID	numeric(18, 0)	

Uses

Name
IPCLOGTRANS_UPDATE_REVERSE_REFERENCE
IPCLOGTRANS

Script

```
-- =====
-- Author: Admin
-- Create date: 14/02/2009
-- Description: Update Log Transaction Info
-- =====
CREATE procedure [dbo].[IPCLOGTRANS_UPDATE_REVERSE_REFERENCE]
    @IPCTRANSID           NUMERIC(18, 0)

AS
BEGIN
    SET DATEFORMAT DMY
    if exists(select * from IPCLOGTRANS where CHAR57='REFU' AND IPCTRANSID=@IPCTRANSID)
        BEGIN
            DECLARE @INSTRUCTIONID AS VARCHAR(50)
            DECLARE @REFID AS VARCHAR(50)
            DECLARE @IPCTRANCODE AS VARCHAR(50)
            SELECT TOP 1 @IPCTRANCODE=IPCTRANCODE, @INSTRUCTIONID=LTRIM(RTRIM(CASE WHEN CHARINDEX('/',CHAR58,1)>0
                THEN SUBSTRING(REPLACE(CHAR58,'REVERSING ',''),1,CHARINDEX('/',REPLACE(CHAR58,'REVERSING ',''),1)-1)
                ELSE
                    END))
                FROM IPCLOGTRANS where
                CHAR57='REFU' AND IPCTRANSID=@IPCTRANSID
            SELECT TOP 1 @REFID=CHAR64 FROM IPCLOGTRANS WHERE IPCTRANCODE= CASE WHEN @IPCTRANCODE='FAST_OUTGOING'
                THEN 'FAST_INCOMING'
                ELSE 'FAST_OUTGOING'
                END
            AND CHARINDEX(@INSTRUCTIONID,CHAR64,1)=CHARINDEX('/',CHAR64,1)+1
            UPDATE IPCLOGTRANS SET CHAR70=ISNULL(@REFID,'') where CHAR57='REFU' AND IPCTRANSID=@IPCTRANSID
        END
    END
```

5.2.48. Procedure: IPCLOGTRANS_UPDATEAPPROVE

Input/Output

	Name	Data type	Description
@	IPCTRANSID	varchar(50)	
@	IPCWORKDATE	varchar(50)	
@	IPCTRANCODE	varchar(50)	
@	USERID	varchar(50)	
@	NEXTUSERAPP	varchar(50)	
@	STATUS	varchar(10)	
@	APPRSTS	int	

Uses

Name
IPCLOGTRANS_UPDATEAPPROVE
IPCLOGTRANS

Script

```

CREATE procedure [dbo].[IPCLOGTRANS_UPDATEAPPROVE]
    -- Add the parameters for the stored procedure here
    @IPCTRANSID VARCHAR(50), @IPCWORKDATE VARCHAR(50), @IPCTRANCODE VARCHAR(50), @USERID VARCHAR(50),
    @NEXTUSERAPP VARCHAR(50),
    @STATUS VARCHAR(10), @APPRSTS int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SET DATEFORMAT DMY
    UPDATE IPCLOGTRANS SET
        --USERID = CASE WHEN USERID = '' THEN @USERID ELSE USERID END,
        USERCURAPP = @USERID,
        NEXTUSERAPP = @NEXTUSERAPP,
        STATUS = @STATUS,
        APPRSTS = @APPRSTS,
        LISTUSERAPP = CASE WHEN LISTUSERAPP IS NULL THEN @USERID ELSE LISTUSERAPP + '#' + @USERID END
    WHERE (IPCTRANSID = @IPCTRANSID AND IPCTRANCODE = @IPCTRANCODE)
    OR (ISBATCH = 'Y' AND BATCHREF = @IPCTRANSID)

END

```

5.2.49. Procedure: IPCLOGTRANS_UPDATEFATHERTRANS

Input/Output

	Name	Data type	Description
→@	IPCTRANSID	varchar(50)	
→@	STATUS	varchar(20)	
→@	DESCRIPTION	varchar(200)	

Uses

Name
IPCLOGTRANS_UPDATEFATHERTRANS
IPCLOGTRANS
IPCLOGTRANSEVENT

TRIAL

Script

```
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[IPCLOGTRANS_UPDATEFATHERTRANS]
    -- Add the parameters for the stored procedure here
    @IPCTRANSID VARCHAR(50),
    @STATUS VARCHAR(20),
    @DESCRIPTION VARCHAR(200)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    --12.9.2017 auto refun after ack FLFI to Fast
    declare @currentstatus as varchar(20)
    declare @ipctrancode as varchar(20)
    declare @num68 as int
    select @currentstatus=status,@ipctrancode=ipctrancode,@num68=num68 from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID

    --30.8.2017 minhmodify char49
    declare @topline as varchar(200)
    --select @char49= CHAR49,@topline=SUBSTRING(CHAR49,1,case when CHARINDEX(CHAR(13),CHAR49,1)=0 then len(char49) else
    CHARINDEX(CHAR(13),CHAR49,1) end ) from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID
    select @topline=SUBSTRING(CHAR49,1,case when CHARINDEX(CHAR(13),CHAR49,1)=0 then len(char49) else
    CHARINDEX(CHAR(13),CHAR49,1)-1 end ) from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID
    declare @lenchar49 as int
    select @lenchar49=len(char49)  from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID
    if(@lenchar49>500) update IPCLOGTRANS set CHAR49=SUBSTRING(char49,1,500) where IPCTRANSID=@IPCTRANSID

    if exists(select * from IPCLOGTRANSEVENT where IPCTRANSID=@IPCTRANSID)
    begin
        UPDATE IPCLOGTRANSEVENT SET STATUSHIS= STATUS,STATUS=@STATUS,CHAR49='FAST-' +@DESCRIPTION where
        IPCTRANSID=@IPCTRANSID
    end
    -- Insert statements for procedure here

    --UPDATE IPCLOGTRANS SET STATUSHIS= STATUS,STATUS=@STATUS,CHAR49='FAST-' +@DESCRIPTION where
    IPCTRANSID=@IPCTRANSID
    --check if current status='FLFI'-> UPDATE TO REFUN STATUS
    if(@ipctrancode='FAST_INCOMING' AND @currentstatus='FLFI' AND @STATUS='ACSP')
    BEGIN
        SET @STATUS='RFRC'
        -- 27.10.2017 MINH MODIFY THIS WHEN incoming FAILED WILL WAIT TX FROM O9CORE SO NOT NEED TO SYNC TO CORE AGAIN.

        --UPDATE IPCLOGTRANS SET NUM68=2,num67=0,STATUSHIS= STATUS,STATUS=@STATUS,CHAR49= case when 'FAST-' +
        +@DESCRIPTION=@topline then CHAR49
        --
        --case when @topline='' then 'FAST-' +@DESCRIPTION else 'FAST-' +@DESCRIPTION + CHAR(13) + CHAR49 end
        --
        --RETURN
    END

    UPDATE IPCLOGTRANS SET STATUSHIS= STATUS,STATUS=@STATUS,CHAR49= case when 'FAST-' +@DESCRIPTION=@topline then
    CHAR49
        else
        case when @topline='' then 'FAST-' +@DESCRIPTION else 'FAST-' +@DESCRIPTION + CHAR(13) + CHAR49 end
            end where IPCTRANSID=@IPCTRANSID
    if(@STATUS<> 'RFRC')
    begin
        --update sync to fast
        if exists(select * from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID and NUM68=1)
        begin
            UPDATE IPCLOGTRANS SET NUM68=2 where IPCTRANSID=@IPCTRANSID
        end
        --sync to core neu thay doi status
        if exists(select * from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID and STATUSHIS<>@STATUS)
        --if exists(select * from IPCLOGTRANS where IPCTRANSID=@IPCTRANSID and NUM70=0)
        begin
            UPDATE IPCLOGTRANS SET NUM70=1,NUM69=0 where IPCTRANSID=@IPCTRANSID
        end
        end
        else
        begin
            UPDATE IPCLOGTRANS SET NUM68=1,num67=0 where IPCTRANSID=@IPCTRANSID
        end
    END
```

5.2.50. Procedure: IPCLOGTRANS_UPDATETIMEOUT

Input/Output

Name		Data type	Description
@IPCTRANSID	IPCTRANSID	varchar(50)	
@APPRSTS	APPRSTS	varchar(10)	

Uses

Name
IPCLOGTRANS_UPDATETIMEOUT
IPCLOGTRANS

Script

```
CREATE proc [dbo].[IPCLOGTRANS_UPDATETIMEOUT]
    @IPCTRANSID VARCHAR(50),
    @APPRSTS VARCHAR(10)
AS
BEGIN
    BEGIN TRAN TIMEOUT
    BEGIN TRY
        UPDATE IPCLOGTRANS SET STATUS = '2', APPRSTS = CAST(@APPRSTS AS NUMERIC)
        WHERE IPCTRANSID = CAST(@IPCTRANSID AS NUMERIC) AND STATUS <> '1'

        IF @@ERROR = 0
        BEGIN
            COMMIT TRAN TIMEOUT
        END
        ELSE
            ROLLBACK TRAN TIMEOUT
    END TRY
    BEGIN CATCH
        ROLLBACK TRAN TIMEOUT
    END CATCH
END CATCH
END
```

5.2.51. Procedure: IPCLOGTRANSDetail_INSERT

Input/Output

	Name	Data type	Description
@	IPCWORKDATE	varchar(10)	
@	IPCTRANSID	numeric(18, 0)	
@	LOGTYPE	varchar(1)	
@	FIELDNAME1	varchar(50)	
@	FIELDVALUE1	nvarchar(2000)	
@	FIELDNAME2	varchar(50)	
@	FIELDVALUE2	nvarchar(2000)	
@	FIELDNAME3	varchar(50)	
@	FIELDVALUE3	nvarchar(2000)	
@	FIELDNAME4	varchar(50)	
@	FIELDVALUE4	nvarchar(2000)	
@	FIELDNAME5	varchar(50)	
@	FIELDVALUE5	nvarchar(2000)	
@	FIELDNAME6	varchar(50)	
@	FIELDVALUE6	nvarchar(2000)	
@	FIELDNAME7	varchar(50)	
@	FIELDVALUE7	nvarchar(2000)	
@	FIELDNAME8	varchar(50)	
@	FIELDVALUE8	nvarchar(2000)	
@	FIELDNAME9	varchar(50)	
@	FIELDVALUE9	nvarchar(2000)	
@	FIELDNAME10	varchar(50)	
@	FIELDVALUE10	nvarchar(2000)	

Uses

	Name
	IPCLOGTRANSDetail_INSERT
	IPCLOGTRANSDetail

Script

```
-- =====
-- Author: Admin
-- Create date: 14/02/2009
-- Description: Log Transaction Detail Info
-- =====
CREATE procedure [dbo].[IPCLOGTRANSDetail_INSERT]
    @IPCWORKDATE          VARCHAR(10),
    @IPCTRANSID            NUMERIC(18, 0),
    @LOGTYPE                VARCHAR(1),
    @FIELDNAME1              VARCHAR(50)='',
    @FIELDVALUE1             NVARCHAR(2000)='',
    @FIELDNAME2              VARCHAR(50)='',
    @FIELDVALUE2             NVARCHAR(2000)='',
    @FIELDNAME3              VARCHAR(50)='',
    @FIELDVALUE3             NVARCHAR(2000)='',
    @FIELDNAME4              VARCHAR(50)='',
    @FIELDVALUE4             NVARCHAR(2000)='',
    @FIELDNAME5              VARCHAR(50)='',
    @FIELDVALUE5             NVARCHAR(2000)='',
    @FIELDNAME6              VARCHAR(50)='',
    @FIELDVALUE6             NVARCHAR(2000)='',
    @FIELDNAME7              VARCHAR(50)='',
    @FIELDVALUE7             NVARCHAR(2000)='',
    @FIELDNAME8              VARCHAR(50)='',
    @FIELDVALUE8             NVARCHAR(2000)='',
    @FIELDNAME9              VARCHAR(50)='',
    @FIELDVALUE9             NVARCHAR(2000)='',
    @FIELDNAME10             VARCHAR(50)='',
    @FIELDVALUE10            NVARCHAR(2000)='',
```

```

@FIELDNAME5           VARCHAR(50)='',
@FIELDVALUE5          NVARCHAR(2000)='',
@FIELDNAME6           VARCHAR(50)='',
@FIELDVALUE6          NVARCHAR(2000)='',
@FIELDNAME7           VARCHAR(50)='',
@FIELDVALUE7          NVARCHAR(2000)='',
@FIELDNAME8           VARCHAR(50)='',
@FIELDVALUE8          NVARCHAR(2000)='',
@FIELDNAME9           VARCHAR(50)='',
@FIELDVALUE9          NVARCHAR(2000)='',
@FIELDNAME10          VARCHAR(50)='',
@FIELDVALUE10         NVARCHAR(2000)='',
--@FIELDNAME11          VARCHAR(50)='',
--@FIELDVALUE11         NVARCHAR(2000)='',
--@FIELDNAME12          VARCHAR(50)='',
--@FIELDVALUE12         NVARCHAR(2000)='',
--@FIELDNAME13          VARCHAR(50)='',
--@FIELDVALUE13         NVARCHAR(2000)='',
--@FIELDNAME14          VARCHAR(50)='',
--@FIELDVALUE14         NVARCHAR(2000)='',
--@FIELDNAME15          VARCHAR(50)='',
--@FIELDVALUE15         NVARCHAR(2000)='',
--@FIELDNAME16          VARCHAR(50)='',
--@FIELDVALUE16         NVARCHAR(2000)='',
--@FIELDNAME17          VARCHAR(50)='',
--@FIELDVALUE17         NVARCHAR(2000)='',
--@FIELDNAME18          VARCHAR(50)='',
--@FIELDVALUE18         NVARCHAR(2000)='',
--@FIELDNAME19          VARCHAR(50)='',
--@FIELDVALUE19         NVARCHAR(2000)='',
--@FIELDNAME20          VARCHAR(50)='',
--@FIELDVALUE20         NVARCHAR(2000)=''

AS
BEGIN
    SET DATEFORMAT DMY
    IF (@FIELDNAME1 <> '') BEGIN
        INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
        VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME1, @FIELDVALUE1)
    END
    IF (@FIELDNAME2 <> '') BEGIN
        INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
        VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME2, @FIELDVALUE2)
    END
    IF (@FIELDNAME3 <> '') BEGIN
        INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
        VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME3, @FIELDVALUE3)
    END
    IF (@FIELDNAME4 <> '') BEGIN
        INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
        VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME4, @FIELDVALUE4)
    END
    IF (@FIELDNAME5 <> '') BEGIN
        INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
        VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME5, @FIELDVALUE5)
    END
    IF (@FIELDNAME6 <> '') BEGIN
        INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
        VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME6, @FIELDVALUE6)
    END
    IF (@FIELDNAME7 <> '') BEGIN
        INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
        VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME7, @FIELDVALUE7)
    END
    IF (@FIELDNAME8 <> '') BEGIN
        INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
        VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME8, @FIELDVALUE8)
    END
    IF (@FIELDNAME9 <> '') BEGIN
        INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
        VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME9, @FIELDVALUE9)
    END
    IF (@FIELDNAME10 <> '') BEGIN
        INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
        VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME10, @FIELDVALUE10)
    END
    --IF (@FIELDNAME11 <> '') BEGIN
    --    INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
    --    VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME11, @FIELDVALUE11)
    --END
    --IF (@FIELDNAME12 <> '') BEGIN
    --    INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
    --    VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME12, @FIELDVALUE12)
    --END
    --IF (@FIELDNAME13 <> '') BEGIN
    --    INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
    --    VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME13, @FIELDVALUE13)
    --END
    --IF (@FIELDNAME14 <> '') BEGIN
    --    INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
    --    VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME14, @FIELDVALUE14)
    --END
    --IF (@FIELDNAME15 <> '') BEGIN
    --    INSERT INTO IPCLOGTRANSDetail(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)

```

```
--      VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME15, @FIELDVALUE15)
--END
--IF (@FIELDNAME16 <> '') BEGIN
--      INSERT INTO IPCLOGTRANSDETAIL(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
--      VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME16, @FIELDVALUE16)
--END
--IF (@FIELDNAME17 <> '') BEGIN
--      INSERT INTO IPCLOGTRANSDETAIL(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
--      VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME17, @FIELDVALUE17)
--END
--IF (@FIELDNAME18 <> '') BEGIN
--      INSERT INTO IPCLOGTRANSDETAIL(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
--      VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME18, @FIELDVALUE18)
--END
--IF (@FIELDNAME19 <> '') BEGIN
--      INSERT INTO IPCLOGTRANSDETAIL(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
--      VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME19, @FIELDVALUE19)
--END
--IF (@FIELDNAME20 <> '') BEGIN
--      INSERT INTO IPCLOGTRANSDETAIL(IPCWORKDATE, IPCTRANSID, LOGTYPE, FIELDNAME, FIELDVALUE)
--      VALUES (@IPCWORKDATE, @IPCTRANSID, @LOGTYPE, @FIELDNAME20, @FIELDVALUE20)
--END
END
```

TRIAL

5.2.52. Procedure: IPCLOGTRANSDetail_SELECT

Input/Output

Name		Data type	Description
@IPCWORKDATE		varchar(10)	
@IPCTRANSID		numeric(18, 0)	
@LOGTYPE		varchar(1)	

Uses

Name
IPCLOGTRANSDetail_SELECT
IPCLOGTRANSDetail

Script

```
-- =====
-- Author: Admin
-- Create date: 14/02/2009
-- Description: Select Transaction Detail
-- =====
CREATE procedure [dbo].[IPCLOGTRANSDetail_SELECT]
    @IPCWORKDATE      VARCHAR(10),
    @IPCTRANSID       NUMERIC(18,0),
    @LOGTYPE          VARCHAR(1)
AS
BEGIN
    SET DATEFORMAT DMY
    SELECT A.LOGTYPE, A.FIELDNAME, A.FIELDVALUE
    FROM IPCLOGTRANSDetail A (NOLOCK)
    WHERE IPCWORKDATE = @IPCWORKDATE
        AND IPCTRANSID = @IPCTRANSID AND LOGTYPE = @LOGTYPE
END
```

5.2.53. Procedure: IPCLOGTRANSDetailFIELD_SELECT

Input/Output

	Name	Data type	Description
→@	SOURCEID	varchar(20)	
→@	DESTID	varchar(20)	
→@	IPCTRANCODE	varchar(20)	
→@	LOGTYPE	varchar(1)	

Uses

Name
IPCLOGTRANSDetailFIELD_SELECT
IPCLOGTRANSDetailFIELD

Script

```
-- =====
-- Author: Admin
-- Create date: 14/02/2009
-- Description: Select Field Transaction Detail
-- =====
CREATE procedure [dbo].[IPCLOGTRANSDetailFIELD_SELECT]
    @SOURCEID           VARCHAR(20),
    @DESTID             VARCHAR(20),
    @IPCTRANCODE        VARCHAR(20),
    @LOGTYPE            VARCHAR(1)
AS
BEGIN
    SELECT FIELDNAME FROM IPCLOGTRANSDetailFIELD A (NOLOCK)
    WHERE A.SOURCEID = @SOURCEID AND (A.DESTID = @DESTID OR @DESTID = '')
        AND A.IPCTRANCODE = @IPCTRANCODE AND A.LOGTYPE = @LOGTYPE
END
```

5.2.54. Procedure: IPCSCHEDULE_UPDATENEXTEXEC

Input/Output

	Name	Data type	Description
→@	SCHEDULEID	varchar(100)	
→@	NEXTEXEC	datetime	

Uses

Name
IPCSCHEDELE_UPDATENEXTEXEC
IPC_SCHEDULES (Schedule Information)

Script

```
CREATE PROCEDURE [dbo].[IPCSCHEDULE_UPDATENEXTEXEC]
@SCHEDULEID VARCHAR(100),
@NEXTEXEC DATETIME
AS
BEGIN
    UPDATE dbo.IPC_SCHEDULES SET NEXTEXECUTE=DATEADD(SECOND,3, GETDATE()) WHERE SCHEDULEID=@SCHEDULEID
END
```

5.2.55. Procedure: IPCSYSVAR_INSERTO9SESSION

Input/Output

Name	Data type	Description
@VARVALUE	nvarchar(4000)	

Uses

Name
IPCSYSVAR_INSERTO9SESSION
IPCSYSVAR

Script

```
CREATE procedure [dbo].[IPCSYSVAR_INSERTO9SESSION]
    @VARVALUE NVARCHAR(4000)
AS
BEGIN
IF (@VARVALUE <>'False' AND @VARVALUE <> 'STRDESTRESULT' AND @VARVALUE <> '')
BEGIN
    IF NOT EXISTS (SELECT * FROM dbo.IPCSYSVAR WHERE VARNAME='O9SESSION')
        INSERT INTO dbo.IPCSYSVAR(VARNAME,VARVALUE) VALUES ('O9SESSION',@VARVALUE)
    ELSE
        UPDATE IPCSYSVAR SET VARVALUE = @VARVALUE WHERE VARNAME = 'O9SESSION'
END
END
```

TRIAL

5.2.56. Procedure: IPCSYSVAR_SELECT

Uses

Name
IPCSYSVAR_SELECT
IPCSYSVAR

Script

```
-- =====
-- Author:      Admin
-- Create date: 14/02/2009
-- Description: Check Exist SourceTranRef
-- =====
CREATE procedure [dbo].[IPCSYSVAR_SELECT]

AS
BEGIN
    SELECT *
        FROM IPCSYSVAR A (NOLOCK)
    WHERE VARNAME != 'SYSTEMSTATUS'

END
```

TRIAL

5.2.57. Procedure: IPCSYSVAR_UPDATE

Input/Output

	Name	Data type	Description
@	VARNAME	varchar(50)	
@	VARVALUE	nvarchar(4000)	

Uses

Name
IPCSYSVAR_UPDATE
IPCSYSVAR

Script

```
-- =====
-- Author:          Admin
-- Create date:    14/02/2009
-- Description:     Update Sysvar
-- =====
CREATE procedure [dbo].[IPCSYSVAR_UPDATE]
    @VARNAME  VARCHAR(50),
    @VARVALUE NVARCHAR(4000)
AS
BEGIN
    UPDATE IPCSYSVAR SET VARVALUE = @VARVALUE WHERE VARNAME = @VARNAME
END
```

5.2.58. Procedure: PAGES_CHECKPERMIT

Input/Output

	Name	Data type	Description
@PAGEID	PAGEID	int	
@USERNAME	USERNAME	varchar(50)	
@SERVICEID	SERVICEID	varchar(50)	

Uses

Name
PAGES_CHECKPERMIT
EBA_TRANRIGHT
SMP_PAGES
SMP_USERINROLE

Script

```

create PROCEDURE [dbo].[PAGES_CHECKPERMIT]
    -- ADD THE PARAMETERS FOR THE STORED PROCEDURE HERE
    @PAGEID INT,
    @USERNAME VARCHAR(50),
    @SERVICEID VARCHAR(50)

AS
BEGIN
    -- SET NOCOUNT ON ADDED TO PREVENT EXTRA RESULT SETS FROM
    -- INTERFERING WITH SELECT STATEMENTS.
    SET NOCOUNT ON;
    IF NOT EXISTS (SELECT * FROM SMP_USERINROLE WHERE USERNAME='GUEST' AND ROLEID=1)
        INSERT INTO SMP_USERINROLE SELECT 'GUEST',1
    -- INSERT STATEMENTS FOR PROCEDURE HERE
    IF(@SERVICEID='SEMS' OR @SERVICEID='SMP')
        BEGIN
            SELECT B.ROLEID
            FROM SMP_PAGES A INNER JOIN EBA_TRANRIGHT B ON A.PAGEID=B.PAGEID
                INNER JOIN SMP_USERINROLE C ON B.ROLEID=C.ROLEID
            WHERE A.ISSHOW=1 AND C.USERNAME=@USERNAME AND A.PAGEID=@PAGEID
        END
END

```

5.2.59. Procedure: Pages_GetPageInfo

Input/Output

Name	Data type	Description
@pageid	int	

Uses

Name
Pages_GetPageInfo
SMP_masterpage
SMP_PAGES
SMP_theme

Script

```
-- =====
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
create procedure [dbo].[Pages_GetPageInfo]
    -- Add the parameters for the stored procedure here
    @pageid int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT * FROM SMP_PAGES a (nolock) inner join SMP_masterpage b on a.masterpageid=b.masterpageid
    inner join SMP_theme c on a.theme=c.themeid
    WHERE PAGEID=@pageid
END
```

5.2.60. Procedure: POSITION_LOAD

Input/Output

Name	Data type	Description
@PAGEID	int	

Uses

Name
POSITION_LOAD
SMP_PAGES
SMP_POSITION

Script

```
create PROCEDURE [dbo].[POSITION_LOAD]
    @PAGEID INT
AS
BEGIN
    SET NOCOUNT ON;
    SELECT A.* FROM SMP_POSITION A INNER JOIN SMP_PAGES B ON A.MASTERPAGEID=B.MASTERPAGEID
    WHERE B.PAGEID=@PAGEID
END
```

5.2.61. Procedure: Settings_LoadPortalSettings

Uses

Name
 Settings_LoadPortalSettings
 SMP_Settings

Script

```
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
CREATE procedure [dbo].[Settings_LoadPortalSettings]
    -- Add the parameters for the stored procedure here
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT * from SMP_Settings
END
```

TRIAL

5.2.62. Procedure: USERS_GETROLE

Input/Output

Name		Data type	Description
@USERNAME		nvarchar(50)	
@SERVICEID		varchar(50)	

Uses

Name
USERS_GETROLE
EBA_ROLES
SMP_USERINROLE
SMP_USERS

Script

```
-- =====
-- AUTHOR: <AUTHOR,,NAME>
-- CREATE DATE: <CREATE DATE,,>
-- DESCRIPTION: <DESCRIPTION,,>
-- =====
create PROCEDURE [dbo].[USERS_GETROLE]
    -- ADD THE PARAMETERS FOR THE STORED PROCEDURE HERE
    @USERNAME NVARCHAR(50),
    @SERVICEID VARCHAR(50)
AS
BEGIN
    -- SET NOCOUNT ON ADDED TO PREVENT EXTRA RESULT SETS FROM
    -- INTERFERING WITH SELECT STATEMENTS.
    SET NOCOUNT ON;

    -- INSERT STATEMENTS FOR PROCEDURE HERE
    IF(@SERVICEID='SEMS' OR @SERVICEID='SMP')
    BEGIN
        SELECT A.* ,C.ROLEID,C.ROLENAME,C.ROLEDESCRIPTION FROM SMP_USERS A INNER JOIN SMP_USERINROLE B ON
A.USERNAME=B.USERNAME
        INNER JOIN EBA_ROLES C ON B.ROLEID=C.ROLEID
        WHERE A.USERNAME=@USERNAME
    END
END
```

5.2.63. Procedure: WIDGETRIGHT_GETMODULEINPAGEFORROLE

Input/Output

	Name	Data type	Description
@	PORTALID	int	
@	PAGEID	int	
@	USERNAME	varchar(50)	
@	LANG	varchar(50)	
@	SERVICEID	varchar(50)	

Uses

Name
WIDGETRIGHT_GETMODULEINPAGEFORROLE
SMP_USERINROLE
SMP_WIDGETPAGE
SMP_WIDGETRIGHT
SMP_WIDGETS
SMP_WIDGETSLANG

Script

```

create PROCEDURE [dbo].[WIDGETRIGHT_GETMODULEINPAGEFORROLE]
    @PORTALID INT,
    @PAGEID INT,
    @USERNAME VARCHAR(50),
    @LANG VARCHAR(50),
    @SERVICEID VARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;
    IF (@SERVICEID='SMP' OR @SERVICEID='SEMS')
    BEGIN
        SELECT DISTINCT A.* ,B.WIDGETID,C.WIDGETTITLE,B.WIDGETLINK,B.ICONPATH,B.ENABLETHEME,B.SHOWTITLE
        FROM SMP_WIDGETPAGE A INNER JOIN SMP_WIDGETS B ON A.WIDGETID=B.WIDGETID
        INNER JOIN SMP_WIDGETSLANG C ON B.WIDGETID=C.WIDGETID
        INNER JOIN SMP_WIDGETRIGHT D ON A.WIDGETPAGEID=D.WIDGETPAGEID
        INNER JOIN SMP_USERINROLE E ON D.ROLEID=E.ROLEID
        WHERE PORTALID=@PORTALID
        AND PAGEID=@PAGEID AND E.USERNAME=@USERNAME AND LANGID=@LANG AND B.ISPUBLISH=1
        ORDER BY [ORDER],A.WIDGETID
    END
END

```

5.2.64. Procedure: WIDGETS_LOAD

Input/Output

	Name	Data type	Description
@	LANGID	varchar(50)	
@	KEYWORD	nvarchar(20)	

Uses

Name
WIDGETS_LOAD
SMP_WIDGETS
SMP_WIDGETSLANG

Script

```
create PROCEDURE [dbo].[WIDGETS_LOAD]
    @LANGID VARCHAR(50),
    @KEYWORD NVARCHAR(20)
AS
BEGIN
    SET NOCOUNT ON;
    IF (@KEYWORD=' ')
        BEGIN
            SELECT * FROM SMP_WIDGETS A INNER JOIN SMP_WIDGETSLANG B ON A.WIDGETID=B.WIDGETID
            WHERE LANGID=@LANGID
            ORDER BY B.WIDGETTITLE
        END
    ELSE
        BEGIN
            SELECT * FROM SMP_WIDGETS A INNER JOIN SMP_WIDGETSLANG B ON A.WIDGETID=B.WIDGETID
            WHERE LANGID=@LANGID AND B.WIDGETTITLE LIKE N'%' +@KEYWORD+'%'
            ORDER BY B.WIDGETTITLE
        END
END
```

5.3. Functions

5.3.1. Function: fn_ParseText2Table

Input/Output

	Name	Data type	Description
@	Returns	table type	
@	p_SourceText	varchar(8000)	
@	p_Delimeter	varchar(100)	

Script

```
CREATE function[dbo].[fn_ParseText2Table]
(
@p_SourceText  varchar(8000)
,@p_Delimeter varchar(100) = ',' --default to comma delimited.
)
RETURNS @retTable TABLE
(
    Position int identity(1,1)
    ,Int_Value int
    ,Num_value Numeric(18,3)
    ,txt_value varchar(2000)
)
AS
/*
*****
Purpose: Parse values from a delimited string
& return the result as an indexed table
Copyright 1996, 1997, 2000, 2003 Clayton Groom (Clayton_Groom@hotmail.com)
Posted to the public domain Aug, 2004
06-17-03 Rewritten as SQL 2000 function.
Reworked to allow for delimiters > 1 character in length
and to convert Text values to numbers
*****
*/
BEGIN
DECLARE @w_Continue int
,@w_StartPos int
,@w_Length int
,@w_Delimeter_pos int
,@w_tmp_int int
,@w_tmp_num numeric(18,3)
,@w_tmp_txt varchar(2000)
,@w_Delimeter_Len tinyint
if len(@p_SourceText) = 0
begin
    SET @w_Continue = 0 -- force early exit
end
else
begin
    -- parse the original @p_SourceText array into a temp table
    SET @w_Continue = 1
    SET @w_StartPos = 1
    SET @p_SourceText = RTRIM( LTRIM( @p_SourceText))
    SET @w_Length = DATALENGTH( RTRIM( LTRIM( @p_SourceText)))
    SET @w_Delimeter_Len = len(@p_Delimeter)
end
WHILE @w_Continue = 1
BEGIN
    BEGIN
        SET @w_Delimeter_pos = CHARINDEX( @p_Delimeter
            ,(SUBSTRING( @p_SourceText, @w_StartPos
            ,((@w_Length - @w_StartPos) + @w_Delimeter_Len)))
        )
        IF @w_Delimeter_pos > 0 -- delimiter(s) found, get the value
        BEGIN
            SET @w_tmp_txt = LTRIM(RTRIM( SUBSTRING( @p_SourceText, @w_StartPos
            ,(@w_Delimeter_pos - 1)) ))
            if isnumeric(@w_tmp_txt) = 1
            begin
                set @w_tmp_int = cast( cast(@w_tmp_txt as numeric) as int)
                set @w_tmp_num = cast( @w_tmp_txt as numeric(18,3))
            end
            else
            begin
                set @w_tmp_int = null
                set @w_tmp_num = null
            end
            SET @w_StartPos = @w_Delimeter_pos + @w_StartPos + (@w_Delimeter_Len- 1)
        END
    END
END
```

```
ELSE -- No more delimiters, get last value
BEGIN
  SET @w_tmp_txt = LTRIM(RTRIM( SUBSTRING( @p_SourceText, @w_StartPos
    ,((@w_Length - @w_StartPos) + @w_Delimeter_Len)) ))
  if isnumeric(@w_tmp_txt) = 1
  begin
    set @w_tmp_int = cast( cast(@w_tmp_txt as numeric) as int)
    set @w_tmp_num = cast( @w_tmp_txt as numeric(18,3))
  end
  else
  begin
    set @w_tmp_int = null
    set @w_tmp_num = null
  end
  SELECT @w_Continue = 0
END
INSERT INTO @retTable VALUES( @w_tmp_int, @w_tmp_num, @w_tmp_txt )
END
RETURN
END
```

TRIAL

5.3.2. Function: Split

Input/Output

	Name	Data type	Description
@return	Returns	table type	
>@	InputString	varchar(8000)	
>@	Delimiter	varchar(50)	

Used By

Name
fx Split
FAST_RESEND_TX
IPCLOGTRANS_GETTRANSACTION_TOFAST (Get transaction send to FAST)

Script

```

CREATE FUNCTION [dbo].[Split]
    (@InputString           VARCHAR(8000),
     @Delimiter              VARCHAR(50))
)
RETURNS @Items TABLE (
    Item                  VARCHAR(8000)
)
AS
BEGIN
    IF @Delimiter = ''
    BEGIN
        SET @Delimiter = ','
        SET @InputString = REPLACE(@InputString, ' ', @Delimiter)
    END

    IF (@Delimiter IS NULL OR @Delimiter = '')
        SET @Delimiter = ','

--INSERT INTO @Items VALUES (@Delimiter) -- Diagnostic
--INSERT INTO @Items VALUES (@InputString) -- Diagnostic

    DECLARE @Item          VARCHAR(8000)
    DECLARE @ItemList       VARCHAR(8000)
    DECLARE @DelimIndex     INT

    SET @ItemList = @InputString
    SET @DelimIndex = CHARINDEX(@Delimiter, @ItemList, 0)
    WHILE (@DelimIndex != 0)
    BEGIN
        SET @Item = SUBSTRING(@ItemList, 0, @DelimIndex)
        INSERT INTO @Items VALUES (@Item)

        -- Set @ItemList = @ItemList minus one less item
        SET @ItemList = SUBSTRING(@ItemList, @DelimIndex+1, LEN(@ItemList)-@DelimIndex)
        SET @DelimIndex = CHARINDEX(@Delimiter, @ItemList, 0)
    END -- End WHILE

    IF @Item IS NOT NULL -- At least one delimiter was encountered in @InputString
    BEGIN
        SET @Item = @ItemList
        INSERT INTO @Items VALUES (@Item)
    END

    -- No delimiters were encountered in @InputString, so just return @InputString
    ELSE INSERT INTO @Items VALUES (@InputString)

    RETURN
END -- End Function

```