IBM i

Naming Conventions

IBM i Center of Excellence (iCOE)

Software Design Document

General Information	
Product Name and Version	IBM i, Iseries, AS/400 Software Development v2020.05
Feature	ILE, SQL and miscellaneous Naming Conventions
Main Contact	Hiram Hubbard – IBM i Center of Excellence
Author(s)	Hiram Hubbard, IBM i Senior Solutions Architect
Created Date	August 10, 2020
Last Updated by	Hiram Hubbard, IBM i Senior Solutions Architect
Last Updated on	January 4, 2022

1 Purpose

The purpose of this document is to define the structure and naming conventions to be used for the development of all ILE and SQL objects on IBM i. This document serves as a guideline for all Software development on IBM i, AS/400, iSeries. The main goal of adopting a naming convention for software development objects is so that you and others can easily identify the type and purpose of all objects contained in the system. The information presented here serves as a guide for you to follow when naming your objects. When reading these rules and guidelines, remember that consistent naming can be the most important rule to follow. The goal of naming convention is to produce practical, legible, concise, unambiguous and consistent names for your objects.

2 ILE Naming Conventions

Type	**Template**	**Description**
RPG Application Name	**xx R nnn A s**	***xx*** is a two character prefix for the Module, ***R*** is for RPG, ***nnn*** is for sequence number (001 - 999) , ***A*** for Application and ***s*** is the Index (1 - 9). ***example: APR120A, GLR150A1***
CLLE Application Program	**xx C nnn A s**	***xx*** is a two character prefix for the Module, ***C*** is for CLLE, *nnn* is for sequence number (001 - 999) , ***A*** for Application and **s** is the Index (1 - 9). ***example: APC120A, IMC020A1***
Display File	**xx D nnn FM s**	*xx* is a two character prefix for the Module, ***D*** is for Display File, *nnn* is for sequence number (001 - 999) , ***FM*** for Screen Format and **s** is the Index (1 - 9).
Record Format	**xx F nnn S x**	***xx*** is a two character prefix for the Module, ***F*** is for format , ***nnn*** is for sequence number (001 - 999) , **s** for Screen Format and **x** is the Index (1 - 9).
RPG *Module	**xx R nnn M s**	***xx*** is a two character prefix for the Module, ***R*** is for RPG, ***nnn*** is for sequence number (001 - 999) , ***M*** for *Module and **s** is the Index (1 - 9). ***example: SAR100M***
CLLE *Module	**xx C nnn M s**	***xx*** is a two character prefix for the Module, ***C*** is for CLLE, ***nnn*** is for sequence number (001 - 999) , ***M*** for *Module and **s** is the Index (1 - 9).
RPG /CopyBook Prototype	**xx R nnn P s**	**xx** is a two character prefix for the Module, ***R*** is for RPG, ***nnn*** is for sequence number (001 - 999) , ***P*** for Prototype /Copybook and ***s*** is the Index (1 - 9). ***example: PMR100P,***

*SRVPGM Service Program	xx T nnn S s	***xx*** is a two character prefix for the Module, ***T*** is for Type (R=RPG and C=CLLE) , **nnn** is for sequence number (001 - 999) , **S** for *SRVPGM and **s** is the Index (1 - 9).
Externally-described data structure	xx S nnn DS	***xx*** is a two character prefix for the Module, ***S*** is for Table, ***nnn*** is for sequence number (001 - 999) and ***DS*** for Data Structure.
RPG *Module *Pgm Call	xx R nnn M Ps	***xx*** is a two character prefix for the Module, ***R*** is for RPG, ***nnn*** is for sequence number (001 - 999) and ***M*** for *Module, **s** is the Index (1 -9) and ***P*** for *PGM of the *Module (NO Source).
CLLE *Module *Pgm Call	xx C nnn M Ps	***xx*** is a two character prefix for the Module, ***C*** is for CLLE, ***nnn*** is for sequence number (001 - 999) and ***M*** for *Module, **s** is the Index (1 -9) and ***P*** for *PGM of the *Module (NO Source).
Subprocedures	camelCase	Camel case may start with a capital character or with a lowercase character. Common examples include: **"iPhone", "macOS", "eCommerce", "gEtNextPoNumber" or "doBLProcess**".
Subroutine	SR_camelCase	**SR_** is a three character prefix for Subroutine , Camel case may start with a capital character or with a lowercase character. Examples: **"*SR_IOGetData*", "*SR_cHeckInput*".** ***Rerfer to Guideline***
Binding Source Member	xx nnn_BND	**xx** is a two character prefix for the Module, **nnn** is for sequence number (001 - 999) and ***_BND*** for Bind Source Refer to Guideline
Binding Directory	xxxx BNDDIR	**xxxx** is a four character prefix for the Module and BNDDIR for Binding Directory. *Refer to Guideline*

Stand along Program Variables	**v_nnnnn**	***v_*** is a the prefix for the variable and ***nnnnn*** for long description variable name. example: ***v_sKu, v_nUmberOfRecords, v_ReturnCode***
Global Program Variables	**gv_nnnnn**	***gv_*** is a the prefix for the global variable and ***nnnnn*** for long description variable name. example: ***gv_sKu, gv_nUmberOfRecords, gv_ReturnCode***
/Copybook Prototype Variables	**vvv_nnnnn**	***#xx_*** is a four character prefix for the Module variable name and **nnnnn** for long description variable name. example: ***#Pm_nUmberOfRecords, #Iv_ReturnCode, #Cf_Sku***
Screen Display Variables	**@nnnnn**	@ is the prefix for the screen variable and ***nnnnn*** for short *abbreviated* description name. example: ***@sKu, @Store, and @Price***
Service Program Signature	**MODxxxxx.vnmm**	***MOD*** is a three character prefix for the Modification ID, ***xxxxx*** is for modification number, ***.vn*** is for version number and **mm** for version mod number (1 - 99) example (***MOD90004.v1005***) *Refer to Guideline*
Name Activation Group	**nnnnn**	***xxxxx*** is a 10-character name for activation group. *Refer to Guideline* ***example: ITEXTTOOL***
Name Indicator	**nnnnn**	***nnnnn*** is a 30 characters or less name description for the indicator. ***example: ExitKey, Bottom, ClearOutSubfile***, etc.

3 DB2 for i SQL Naming Conventions

Please use the following as a guideline for all DB2 for i objects long name

- Try to limit the name to 50 characters (shorter is better)
- Use underscores for word separation
- Use a letter as the first character of the name. (don't start names with underscores or any special character or number.)
- Limit the use of abbreviations (can lead to misinterpretation of names)
- Limit the use of acronyms (some acronyms have more than one meaning e.g. "ASP")
- Make the name readable (they shouldn't sound funny when read aloud)

TABLES, INDEXES, VIEWS and MQT's

When naming your database objects, give consideration to other steps in the development process. We create long and short database names for new SQL source. Keep in mind you will most likely have to utilize the names you give your tables several times as part of other objects, for example, procedures, triggers or UDF's may all contain references to the table name. You want to keep the name practical, legible, concise, unambiguous and consistent names.

Type	**Template**	**Description**
SQL Table System Object Name	**xx S nnn T s**	***xx*** is a two character prefix for the Module, ***S*** is for SQL Table, ***nnn*** is for sequence number (001 - 999) , ***T*** for Type of table (***T*** = Transaction or Audit, ***M*** = Master, ***W*** = Work, ***Q*** = Temp) and ***s*** is the Index (1 - 9). *Same as member* ***example: APS100T, GLS120M, INS200W or PMS150Q1***
SQL Table Long Name	**xx_ nnnn _xxx**	***xx_*** is a three character prefix for the Module, ***nnnn*** is for long description name and ***_xxx*** for IBM i source member ***example: PM_Price_Master_PMS100M or SA_Sales_Audit_SAS100T***
SQL Index System Object Name	**xx I nnn T s**	***xx*** is a two character prefix for the Module, ***I*** is for SQL Index, ***nnn*** is for sequence number (001 - 999) , ***T*** for Type of Index (***E*** = Encoder Vector, ***I*** = Default) and ***s*** is the Index (1 - 9). *Same as member* ***example PMI150E1 or SAI200I5***
SQL Index Long Name	**xx_ nnnn _xxx**	***xx_*** is a three character prefix for the Module and ***nnnn*** is for long description name and ***_xxx*** for IBM i source member ***example: PM_ Price_Master_PMI100E or IN_Inventory_Audit_INI100E1***
SQL View System Object Name	**xx t nnn T s**	***xx*** is a two character prefix for the Module, ***V*** is for SQL View, ***nnn*** is for sequence number (001 - 999) , ***T*** for Type of View (***V*** = Single view, ***VV*** = View of View) and ***s*** is the Index (1 - 9). *Same as member* ***example: APV100V1, GLV120V, INV200V or PMV150VV***

SQL View Long Name	**xx_ nnnn _xxx**	***xx_*** is a three character prefix for the Module, ***nnnn*** is for long description name and ***_xxx*** for IBM i source member ***example: PM_Price_Changes_PMV100VV or PO_Receipt_Audit_INV100V1***
SQL MQT System Object Name	**MQT**	***For all MQT’s please refer to the MQT Standards Doc***
SQL MQT Long Name	**MQT**	***For all MQT’s please refer to the MQT Standards Doc***
SQL Control for Permission Source member	**xx X nnn P s**	***xx*** is a two character prefix for the Module, ***X*** is for control, ***nnn*** is for sequence number (001 - 999) ***P*** for permission and ***s*** is the Index (1 - 99). *Same as member* ***example: TAX100P or SAX250P1*** ***Please Note: CONTROL’s can only be created by iCOE.***
SQL Control for Permission Long Name	**xx_ PERMISSION_ nnnn _xxx**	***xx_*** is a three character prefix for the Module, ***PERMISSION_*** is control type, ***nnnn*** is for the object table name and ***_xxx*** for IBM i source member ***example: HR_PERMISSION_TAS100M_TAX100P***

Type	Template	Description
SQL Control for Mask Source member	**xx X nnn M s**	***xx*** is a two character prefix for the Module, ***X*** is for control, ***nnn*** is for sequence number (001 - 999) ***M*** for mask and ***s*** is the Index (1 - 99). *Same as member* ***example: TAX100M or HRX250M1*** ***Please Note: CONTROL's can only be created by iCOE.***
SQL Control for Mask Long Name	**xx_ MASK_ nnnn _yyy _xxx**	***xx_*** is a three character prefix for the Module, ***MASK_*** is control type, ***nnnn*** is for the object table name, and ***_yyy*** for column name and ***_xxx*** for IBM i source member ***example: HR_MASK_TAS100M_ESSN_TAX100M***
SQL Type ARRAY	**xx E nnn ARY s**	***xx*** is a two character prefix for the Module, ***E*** is for SQL Type, **nnn** is for sequence number (001 - 999) , ***ARY*** for Array suffix and **s** is the Index (1 - 9). ***example: PME100ARY or INE250ARY1*** ***Note: Object and Source must match.***

Type	Template	Description
SQL Stored Procedure System Object Name	**xx E nnn PP s**	***xx*** is a two character prefix for the Module, ***E*** is for SQL Type, ***nnn*** is for sequence number (001 - 999) ***PP*** for procedure type (***AP*** = Add Procedure, ***BP*** = Browse Procedure, ***DP*** = Delete Procedure, ***UP*** = Update Procedure or ***MP*** = Multi Procedure) and ***s*** is the Index (1 - 99). *Same as member* ***example: ITE280AP, ITE284BP, PME490MP or INE100UP1***
SQL Stored Procedure Long Name	**USP _nnn _ yyy _xxx**	***UPS*** is a three character prefix for the User Stored procedure, ***_nnn*** is for function (***Insert, Update, Delete, Select or Multi***) ***_yyy*** for table name or label and ***_xxx*** for IBM i source member. ***example: USP_SELECT_ITS281T_ITE281BP or*** ***USP_SELECT_DYNAMIC_SEE901BP***
SQL Trigger System Object Name	**xx G nnn PP s**	***xx*** is a two character prefix for the Module, ***G*** is for SQL Trigger, ***nnn*** is for sequence number (001 - 999) ***PP*** for Trigger type (***AT*** = Add trigger, ***DT*** = Delete trigger, ***UT*** = Update trigger or ***MT*** = Multi trigger) and ***s*** is the Index (1 - 99). *Same as member* ***example: PMG101AT, ING200MT, GLG120UT***

Type	**Template**	**Description**
SQL Trigger Long Name	**UTG _nnn _ yyy _xxx**	***UTG*** is a three character prefix for the User Trigger, ***_nnn*** is for function (***Insert, Update, Delete, or Multi***) ***_yyy*** for table name or label and ***_xxx*** for IBM i source member. ***example: UTG_INSERT_SAS100M_SAG100AT or*** ***UTG_MULTI_INSVBAL_ING120MT***
SQL Global Variable Object Name	**xx S nnn GV s**	***xx*** is a two character prefix for the Module, ***S*** is for SQL Type, **nnn** is for sequence number (001 - 999) , ***GV*** for Global Variable and **s** is the Index (1 - 9). ***example: PMS100GV or APS250GV1*** ***Note: Object and Source must match.***
SQL Global Variable Long Name	**xx_ VARIABLE nnnn _xxx**	***xx_*** is a three character prefix for the Module, ***VARIABLE_*** is control type, ***nnnn*** is for long description name and ***_xxx*** for IBM i source member ***example: HR_VARIABLE_ APP_ID_TAS100GV or*** ***PM_VARIABLE_ JOB_SESSION_ID_PMS550GV1***