LANSA/AD

Open System Document Internal Database and System Utilities

Release 7.0

The following material is confidential.

This material must not be disclosed in any form to any person other than Employees of ASPECT Computing, Agents acting on their behalf, or any Organisation legally obligated by the signing of a non-disclosure agreement not to disclose or copy its contents.

Copies of this document must be kept in a secure place and no copies may be left with, made by, or kept by, anyone other than Employees of ASPECT Computing, Agents acting on their behalf, or organisations that have signed a non-disclosure agreement.

Contents

Chapter 1. Introduction and Overview

What Is This Guide?

This guide contains technical information about the LANSA internal database and information about the Open Systems Utilities.

Who Should Use This Guide?

This guide should only be used by people who have a sound knowledge of LANSA and its database.

Chapter 2. Installation Procedures

Installing the Open Systems Utilities

To install the Open Systems Utilities do the following.

- Use the RSTLIB command to restore library DC@OPENLIB to whatever you want to call the Open Systems Utilities library.

Example

```
RSTLIB    LIB(DC@OPENLIB)
          DEV( < device > )
          RSTLIB( < osu library name > )
          VOL(*MOUNTED)
```

< device >	Tape device name (IE TAP01).
< osu library name >	Whatever you decide to call this library on your system (IE DC@OPENLIB).

- There should be no errors from this step.
- Open Systems Utilities is now installed on your system.
- Read the rest of this guide before you attempt to use these utilities. This is important as this guide contains instructions on how to actually use these utilities, and warnings about the use of these utilities.

Chapter 3. Internal Database File Layouts

Abbreviations Used in This Section

Abbreviation	Full Meaning
<<pgmlib>>	Name of LANSA program library. Often DC@PGMLIB, but can be changed during installation. Refer to Installation & Maintenance Guide for more details.
<<dtalib>>	Name of LANSA data library. Often DC@DTALIB, but can be changed during installation. Refer to Installation & Maintenance Guide for more details.
ROS	Run only system. A LANSA system that is never used for application development or enhancement.
FFC	Full function checker. A facility used online by developers to check their programs and also used in batch during RDML function compiles.

DC@F01 - Online Help Text Storage

File Name DC@F01

Description Storage of LANSA online help text. Note that this file does not store user help text. File contains the latest version of the LANSA User Guide and LANSA Technical Guide.

Implemented in: 4.00 / D3

Normal Library: <<pgmlib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F01V1	Read	F01PTR,F01SEQ (unique).
DC@F01V2	Update	Same as DC@F01V1.
DC@F01V3	Any	F01TAG, F01PTR, F01SEQ

ROS comments: May be cleared. May also contain multiple members for different development languages.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F01PTR	A	10		Help text internal pointer
F01SEQ	P	4	0	Sequence number
F01ATR	A	1		Field is not currently used.
F01TXT	A	77		Help text line
F01TAG	A	6		Help text tag

DC@F02 - System Security

File Name DC@F02

Description System Security file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F02V1	Read	F02P#I,F02OBJ,F02EXT,F02TYP, F02USR (unique).
DC@F02V2	Update	Same as DC@F02V1.
DC@F02V3	Read	F02P#I,F02USR,F02TYP,F02OBJ, F02EXT (unique).
DC@F02V4	Update	Same as DC@F02V3.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F02OBJ	A	10		Object name
F02EXT	A	10		Object name extension
F02TYP	A	2		Object type
F02USR	A	10		User profile ID
F02LAR	A	20		Access rights list
F02P#I	A	3		Partition identifier

DC@F03 - Field Definition

File Name DC@F03

Description Field Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F03V1	Read	F03P#I,F03FLD (unique).
DC@F03V2	Update	Same as DC@F03V1.
DC@F03V3	Read	F03P#I,F03REF.
DC@F03V4	Update	Same as DC@F03V3.
DC@F03V5	Read	F03P#I,F03ALS.
DC@F03V6	Update	Same as DC@F03V5.
DC@F03V7	Read	F03FLD,F03P#I.
DC@F03V8	Update	Same as DC@F03V7.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F03FLD	A	10		Field name
F03DES	A	40		Field description
F03TYP	A	1		Field type
F03LEN	P	3	0	Field length
F03DEC	P	1	0	Number of decimals
F03LBL	A	15		Field label
F03CH1	A	20		Column heading 1
F03CH2	A	20		Column heading 2
F03CH3	A	20		Column heading 3
F03REF	A	10		Reference name
F03IAT	A	40		Input attributes list
F03OAT	A	40		Output attributes list
F03EDC	A	1		Edit code
F03EDW	A	24		Edit word
F03DFT	A	20		Default value
F03HLP	A	10		Pointer to help text
F03ALS	A	30		Alias name
F03UDT	A	200		Additional information area
F03P#I	A	3		Partition Identifier

DC@F04 - Help Text Storage

File Name DC@F04

Description Help Text Storage file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F04V1	Read	F04P#I,F04HLP,F04SEQ (unique).
DC@F04V2	Update	Same as DC@F04V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F04HLP	A	10		Help text pointer
F04SEQ	P	7	0	Sequence number
F04TYP	A	1		Text type
F04ATR	A	1		Attribute
F04TXT	A	77		Help text
F04P#I	A	3		Partition Identifier

DC@F05 - Validation Check Directory

File Name DC@F05

Description Validation Check Directory file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F05V1	Read	F05P#I,F05FLD,F05SEQ (unique). Select F 05LVL = 'D'
DC@F05V2	Update	Same as DC@F05V1.
DC@F05V3	Read	F05P#I,F05PFL,F05PLB,F05VER F05FLD,F 05SEQ (unique). Select F 05LVL = 'F'
DC@F05V4	Update	Same as DC@F05V3.
DC@F05V5	Read	F05P#I,F05MOD,F05FMT,F05PFL F05PLB,F 05FLD,F05SEQ (unique). Select F 05LVL = 'M'
DC@F05V6	Update	Same as DC@F05V5.
DC@F05V7	Read	F05P#I,F05FLD,F05LVL,F05SEQ
DC@F05V8	Update	Same as DC@F05V7.
DC@F05V9	Read	F05P#I,F05PTR.
DC@F05VA	Update	Same as DC@F05V9.

Name	Use	Keys/Description/Comments
DC@F05VB	Read	F05P#I,F05FLD,F05PFL,F05PLB F05VER,F 05SEQ (unique). Select F 05LVL = 'F' *ANDF05VTP = 'A'
DC@F05VC	Update	Same as DC@F05VB.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F05LVL	A	1		Check level
F05FLD	A	10		Field name
F05PFL	A	10		File name
F05PLB	A	10		File library
F05VER	P	7	0	Version number
F05VTP	A	1		Version type
F05MOD	A	10		Process name
F05FMT	A	7		Function name
F05SEQ	P	15	0	Check sequence no.
F05TYP	A	2		Check type
F05DES	A	30		Check description
F05PAC	A	1		Positive action
F05NAC	A	1		Negative action
F05MSG	A	80		Message text
F05MID	A	7		Message ID
F05MFN	A	10		Message file name
F05MFL	A	10		Message file library
F05PTR	A	10		Pointer to check
F05$AD	A	1		During add ?
F05$CH	A	1		During change ?
F05$DL	A	1		During delete ?
F05P#I	A	3		Partition Identifier

DC@F06 - Range of Values Check

File Name DC@F06

Description Range of Values Check file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F06V1	Read	F06P#I,F06PTR,F06SEQ (unique).
DC@F06V2	Update	Same as DC@F06V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F06PTR	A	10		Pointer to check
F06SEQ	P	3	0	Sequence number
F06LRG	A	20		Low range value
F06HRG	A	20		High range value
F06P#I	A	3		Partition Identifier

DC@F07 - Range of Values "LIST" Check

File Name DC@F07

Description Range of Values "LIST" Check file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F07V1	Read	F07P#I,F07PTR,F07SEQ (unique).
DC@F07V2	Update	Same as DC@F07V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F07PTR	A	10		Pointer to check
F07SEQ	P	3	0	Sequence number
F07VAL	A	20		List value
F07P#I	A	3		Partition Identifier

DC@F08 - Code/File Check

File Name DC@F08

Description Code/File Check file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F08V1	Read	F08P#I,F08PTR (unique).
DC@F08V2	Update	Same as DC@F08V1.
DC@F08V3	Read	F08P#I,F08FIL.
DC@F08V4	Update	Same as DC@F08V3.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F08PTR	A	10		Pointer to check
F08FIL	A	10		File checked against
F08KVL	A	200		List of key values
F08P#I	A	3		Partition Identifier

DC@F09 - Logic Check

File Name DC@F09

Description Simple Logic Check file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F09V1	Read	F09P#I,F09PTR (unique).
DC@F09V2	Update	Same as DC@F09V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F09PTR	A	10		Pointer to check
F09CN1	A	200		Condition part 1
F09CN2	A	200		Condition part 2
F09P#I	A	3		Partition Identifier

DC@F10 - Logic Check

File Name DC@F10

Description Complex Logic Check file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F10V1	Read	F10P#I,F10PTR (unique).
DC@F10V2	Update	Same as DC@F10V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F10PTR	A	10		Pointer to check
F10PGM	A	10		Program to be called
F10APM	A	200		Additional parameters
F10P#I	A	3		Partition Identifier

DC@F11 - Date Range Check

File Name DC@F11

Description Date Range Check file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F11V1	Read	F11P#I,F11PTR (unique).
DC@F11V2	Update	Same as DC@F11V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F11PTR	A	10		Pointer to check
F11DFM	A	8		Date format
F11DBF	P	7	0	Allowable days before
F11DAF	P	7	0	Allowable days after
F11P#I	A	3		Partition Identifier

DC@F12 - File Definition

File Name DC@F12

Description File Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F12V1	Read	F12P#I,F12PFL,F12PLB (unique).
DC@F12V2	Update	Same as DC@F12V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F12PFL	A	10		Physical file name
F12PLB	A	10		File library
F12AVN	P	7	0	Active version no.
F12ATD	P	12	0	Time/Date made active
F12AUS	P	7	0	User who made active
F12UVM	P	7	0	Last version no.
F12P#I	A	3		Partition Identifier

DC@F14 - File Definition "LIST"

File Name DC@F14

Description File Definition "LIST" file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F14V1	Read	F14P#I,F14PTR,F14TYP, F14SEQ (unique).
DC@F14V2	Update	Same as DC@F14V1.
DC@F14V3	Read	F14P#I,F14TYP,F14VAL.
DC@F14V4	Update	Same as DC@F14V3.
DC@F14V5	Read	F14P#I,F14PFL,F14PLB,F14VER, F14TYP,F14VAL.
DC@F14V6	Update	Same as DC@F14V5.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F14PTR	A	10		Pointer value
F14SEQ	P	7	0	Sequence number
F14TYP	A	3		Field type
F14VAL	A	20		List entry
F14AKI	A	20		List entry information
F14PFL	A	10		Physical file name
F14PLB	A	10		Library name
F14VER	P	7	0	Version number
F14P#I	A	3		Partition Identifier

DC@F15 - Logical View Definition

File Name DC@F15

Description Logical File Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F15V1	Read	F15P#I,F15PFL,F15PLB,F15VER F15LGL (unique).
DC@F15V2	Update	Same as DC@F15V1.
DC@F15V3	Read	F15P#I,F15VTP,F15LGL.
DC@F15V4	Update	Same as DC@F15V3.
DC@F15V5	Read	F15P#I,F15LGL.
DC@F15V6	Update	Same as DC@F15V5.
DC@F15V7	Read	F15LGL,F15PLB,F15P#I.
DC@F15V8	Update	Same as DC@F15V7.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F15PFL	A	10		Physical file name
F15PLB	A	10		File library
F15VER	P	7	0	Version no.
F15VTP	A	1		Version type
F15LGL	A	10		Logical view name
F15DES	A	40		Logical file description
F15APM	A	1		Access path maintenance
F15PTR	A	10		Key list pointer
F15AP1	A	200		Add parameters 1
F15AP2	A	200		Add parameters 2
F15SP1	A	1		Spare field 1
F15SP2	A	5		UNIQUE and DYNSLT flags
F15SP3	A	10		Record format name
F15SP4	A	10		Spare field 4
F15SP5	A	20		Spare field 5
F15SP6	P	7	0	Spare field 6
F15SP7	P	7	0	Spare field 7
F15P#I	A	3		Partition Identifier

DC@F16 - Interactive/Batch Exchange

File Name DC@F16

Description Interactive/Batch Exchange file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F16V1	Read	F16RUN,F16SEQ (unique).
DC@F16V2	Update	Same as DC@F16V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F16RUN	P	15	0	Run number
F16SEQ	P	7	0	Sequence number
F16INF	A	200		Exchange information
F16JUL	A	5		Julian date of request

DC@F17 - Foreign Key Check Definition

File Name DC@F17

Description Foreign Key Check Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F17V1	Read	F17P#I,F17PFL,F17PLB,F17VER, F17FCN (unique).
DC@F17V2	Update	Same as DC@F17V1.
DC@F17V3	Read	F17P#I,F17VTP,F17FKF,F17FKL.
DC@F17V4	Update	Same as DC@F17V3.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F17PFL	A	10		Physical file name
F17PLB	A	10		File library
F17VER	P	7	0	Version no.
F17VTP	A	1		Version type
F17FCN	P	3	0	Foreign key check no.
F17DES	A	40		Key description
F17FKF	A	10		Foreign key file
F17FKL	A	10		Foreign key library
F17UAC	A	3		Update action
F17DAC	A	3		Delete action
F17PTR	A	10		Pointer to key list
F17P#I	A	3		Partition Identifier

DC@F18 - Access Route Definition

File Name DC@F18

Description Access Route Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F18V1	Read	F18P#I,F18PFL,F18PLB,F18VER, F18ARN (unique).
DC@F18V2	Update	Same as DC@F18V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F18PFL	A	10		Physical file name
F18PLB	A	10		File library
F18VER	P	7	0	Version no.
F18VTP	A	1		Version type
F18ARN	A	10		Access route name
F18DES	A	40		Access route description
F18ARF	A	10		Access route file
F18ARL	A	10		Access route library
F18MAX	P	7	0	Maximum records expected
F18NRA	A	10		Action to take if no records
F18PTR	A	10		Pointer to key list
F18P#I	A	3		Partition Identifier

DC@F19 - Batch Control Logic Definition

File Name DC@F19

Description Batch Control Logic Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F19V1	Read	F19P#I,F19PFL,F19PLB,F19VER, F19SEQ (unique).
DC@F19V2	Update	Same as DC@F19V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F19PFL	A	10		Physical file name
F19PLB	A	10		File library
F19VER	P	7	0	Version no.
F19VTP	A	1		Version type
F19SEQ	P	7	0	Sequence number
F19DES	A	40		Batch control description
F19PT1	A	10		List of fields pointer
F19BCF	A	10		Batch control file
F19BCL	A	10		Batch control library
F19BSF	A	10		Batch control status
F19PT2	A	10		Pointer to key list
F19P#I	A	3		Partition Identifier

DC@F20 - Process Definition

File Name DC@F20

Description Process Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F20V1	Read	F20P#I,F20MOD (unique).
DC@F20V2	Update	Same as DC@F20V1.
DC@F20V3	Read	F20MOD,F20P#I.
DC@F20V4	Update	Same as DC@F20V3.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F20MOD	A	10		Process name
F20DES	A	40		Process description
F20STS	A	3		Process status
F20EXK	A	2		Exit cmd key
F20MNK	A	2		Menu cmd key
F20DMK	A	2		Dsp msg cmd key
F20ADK	A	2		Add cmd key
F20CHK	A	2		Chg cmd key
F20DLK	A	2		Del cmd key
F20RCL	A	1		RCLRSC flag
F20OVR	A	1		Allow ovr to files
F20HLP	A	10		HELP text pointer
F20MUS	A	1		MENU style
F20R31	A	1		Has DC@F31 records
F20R32	A	1		Has DC@F32 records
F20R44	A	1		Has DC@F44 records
F20P#I	A	3		Partition Identifier

DC@F21 - Function File Usage Cross Reference

File Name DC@F21

Description Function File Usage Cross Reference

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F21V1	Read	F21P#I,F21MOD,F21FMT,F21FIL, F21LIB (unique).
DC@F21V2	Update	Same as DC@F21V1.
DC@F21V3	Read	F21P#I,F21FIL,F21LIB,F21MOD, F21FMT (unique).
DC@F21V4	Update	Same as DC@F21V3.

ROS comments: May be cleared, but cross reference details may be lost.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F21MOD	A	10		Process name
F21FMT	A	7		Function name
F21FIL	A	10		File name
F21LIB	A	10		File Library
F21TYP	A	3		Type
				PHY = Physical via I/O module
				LGL = Logical via I/O module
				PHO = Physical via *DBOPTIMIZE
				LGO = Logical via *DBOPTIMIZE
F21P#I	A	3		Partition Identifier

DC@F22 - Help Text Tags

File Name DC@F22

Description Help Text Tags.

Implemented in: 5.00 / F4

Normal Library: <<pgmlib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F22V1	Read	F22TYP,F22KEY.
DC@F22V2	Update	Same as DC@F22V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F22TYP	A	1		Record type
F22KEY	A	12		Key value 1
F22DTA	A	40		Data value 1

DC@F23 - Function Definition

File Name DC@F23

Description Function Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F23V1	Read	F23P#I,F23MOD,F23FMT (unique).
DC@F23V2	Update	Same as DC@F23V1.
DC@F23V3	Read	F23P#I,F23MOD,F23MSQ,F23FMT.
DC@F23V4	Update	Same as DC@F23V3.
DC@F23V5	Read	F23P#I,F23FMT,F23MOD.
DC@F23V6	Update	Same as DC@F23V5.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F23MOD	A	10		Process name
F23FMT	A	7		Function name
F23DES	A	40		Function description
F23TYP	A	1		Function type
F23STY	A	1		Function status
F23HLP	A	10		HELP text pointer
F23DNF	A	7		Default Next Function
F23ANF	A	140		Allowable next function list
F23FID	A	6		Internal Function ID
F23APM	A	1		Display on Main Menu
F23SP1	A	10		Actual display file name if different to name compiled with after import.
F23SET	P	3	0	Pointer to current DC@F29/30 code set
F23P#I	A	3		Partition Identifier
F23CRL	A	3		Compiled Release Level
F23CPC	A	2		Compiled PC Level
F23MSQ	A	5		Menu Sequence Number
F23SP2	A	3		Spare Field 2
F23SP3	A	5		Spare Field 3
F23SP4	A	10		Spare Field 4

DC@F25 - Function Field Usage Cross Reference

File Name DC@F25

Description Contains a cross reference of the fields referenced by all functions.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F25V1	Read	F25P#I,F25MOD,F25FMT, F25FLD (unique).
DC@F25V2	Update	Same as DC@F25V1.
DC@F25V3	Read	F25P#I,F25FLD,F25MOD.
DC@F25V4	Update	Same as DC@F25V3.

ROS comments: May be cleared, but all cross reference details will be lost.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F25MOD	A	10		Process Name
F25FMT	A	7		Function name
F25FLD	A	20		Field name (DC@F03 or DC@F40).
F25RPG	A	6		RPGIII field name in program
F25P#I	A	3		Partition Identifier

DC@F26 - System Values Storage Directory

File Name DC@F26

Description System Values Storage Directory file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F26V1	Read	F26VAR (unique).
DC@F26V2	Update	Same as DC@F26V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F26VAR	A	20		System variable name
F26DES	A	40		System variable description
F26TYP	A	1		System variable type (A/N)
F26LEN	P	3	0	Length
F26DEC	P	1	0	Number of decimal positions
F26PGM	A	10		Evaluation program name
F26DYN	A	1		Dynamic or static indicator

DC@F27 - File Version Definition

File Name DC@F27

Description File Version Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F27V1	Read	F27P#I,F27PFL,F27PLB, F27VER (unique).
DC@F27V2	Update	Same as DC@F27V1.
DC@F27V3	Read	F27PFL,F27PLB,F27P#I.
DC@F27V4	Update	Same as DC@F27V3.
DC@F27J1	Read	Logical join of DC@F27 and DC@F15 data.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F27PFL	A	10		Physical file name
F27PLB	A	10		File library
F27VER	P	7	0	Version no.
F27VTP	A	1		Version type
F27DES	A	40		File description
F27PTR	A	10		List of fields pointer
F27UUS	A	10		User amendment
F27UTD	P	12	0	Time/Date amendment
F27EXF	A	1		External file indicator
F27AP1	A	200		Additional parameters part 1
F27AP2	A	200		Additional parameters part 2
F27SP1	A	1		Under Commitment Control
F27SP2	A	5		AUTOCOMMIT option
F27SP3	A	10		Record format name
F27SP4	A	10		Spare field 4
F27SP5	A	20		Spare field 5
F27SP6	P	7	0	Spare field 6
F27SP7	P	7	0	Spare field 7
F27P#I	A	3		Partition Identifier

DC@F28 - Master Skeleton Source Member Storage

File Name DC@F28

Description This file stores the master "skeleton" sets of RPG and DDS that are used during I/O module, process and function compiles.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

No Logical Views attached to this file.

ROS comments: Required to compile any object.

Required in RLTB systems unconditionally.

Required in DBCS systems unconditionally.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
SRCSEQ	S	6	2	Source sequence
SRCDAT	S	6	0	Source date
SRCDTA	A	80	0	Source data

DC@F29 - Function Control Commands

File Name DC@F29

Description Function Control Commands file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F29V1	Read	F29P#I,F29MOD,F29FMT,F29SET, F29SEQ (unique).
DC@F29V2	Update	Same as DC@F29V1.

ROS comments: Required. May be cleared to remove all RDML source code from system.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F29MOD	A	10		Process name
F29FMT	A	7		Function name
F29SEQ	P	15	0	Sequence number
F29LBL	A	3		Line label
F29CMD	A	10		Command
F29STS	A	3		Command status
F29SET	P	3	0	Set identifier
F29P#I	A	3		Partition Identifier

DC@F30 - Function Control Parameters

File Name DC@F30

Description Function Control Parameters file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F30V1	Read	F30P#I,F30MOD,F30FMT,F30SET, F30SEQ,F30SQ2 (unique).
DC@F30V2	Update	Same as DC@F30V1.

ROS comments: Required. May be cleared to remove all RDML source code from system.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F30MOD	A	10		Process name
F30FMT	A	7		Function name
F30SEQ	P	15	0	Sequence number
F30SQ2	P	7	0	Sequence number 2
F30PRM	A	55		Parameters in free format
F30SET	P	3	0	Set identifier
F30P#I	A	3		Partition Identifier

DC@F31 - Process Additional Menu Options

File Name DC@F31

Description Process Additional Menu Options file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F31V1	Read	F31P#I,F31MOD,F31SEQ (unique).
DC@F31V2	Update	Same as DC@F31V1.

ROS comments: Required

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F31MOD	A	10		Process name
F31SEQ	P	7	0	Sequence Number
F31DES	A	40		Menu description
F31CMD	A	210		CMD to execute
F31PMT	A	1		Run Time prompt
F31P#I	A	3		Partition Identifier

DC@F32 - Process Parameter Definition

File Name DC@F32

Description Process Parameter Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F32V1	Read	F32P#I,F32MOD,F32PNO (unique).
DC@F32V2	Update	Same as DC@F32V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F32MOD	A	10		Process name
F32PNO	A	2		Parameter Number
F32TYP	A	1		Parameter type
F32LEN	P	3	0	Parameter length
F32DEC	P	1	0	Decimal Positions
F32DES	A	15		Description
F32P#I	A	3		Partition Identifier

DC@F33 - Temporary Work File for Command Processing

File Name DC@F33

Description Temporary Work File for Command Processing

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

No Logical Views attached to this file.

ROS comments: Only required for FFC and compiles.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F33SEQ	P	15	0	Sequence number
F33LBL	A	3		Line label
F33CMD	A	10		Command
$CMDDS	A	4500		Command data

DC@F34 - Temporary Work File in QTEMP for Compiles

File Name DC@F34

Description Temporary Work File in QTEMP for Compiles

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

No Logical Views attached to this file.

ROS comments: Only required for FFC and compiles.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F34SEQ	P	15	0	Sequence number
F34CMD	A	10		Command
F34TYP	A	1		Message type
F34MSG	A	7		Action to take
F34MSF	A	10		Message file name
F34VAR	A	132		Print line / message

DC@F35 - Compiler Work File

File Name DC@F35

Description Compiler Work file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F35V1	Read	F35KV1,F35KV2,F35KV3.
DC@F35V2	Update	Same as DC@F35V1.

ROS comments: Only required for FFC and compiles.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F35KV1	A	10		Key Value 1
F35KV2	A	50		Key Value 2
F35KV3	A	10		Key Value 3
F35DT1	A	500		Data Value 1

DC@F39 - Compiler Work File

File Name DC@F39

Description Compiler Work file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

No Logical Views attached to this file.

ROS comments: Required for FFC and compiles.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F33SEQ	P	15	0	Sequence number
F33LBL	A	3		Line label
F33CMD	A	10		Command
$CMDAT	A	7000		Command attributes
F33P#I	A	3		Partition Identifier

DC@F40 - Process/Function Field Definition

File Name DC@F40

Description Process/Function Field Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F40V1	Read	F40P#I,F40MOD,F40FMT, F40FLD (unique).
DC@F40V2	Update	Same as DC@F40V1.
DC@F40J1	Read	Logical join of DC@F40 and DC@F03 data.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F40MOD	A	10		Process name
F40FMT	A	7		Function name
F40FLD	A	10		Field name
F40DES	A	40		Field description
F40TYP	A	1		Field type
F40LEN	P	3	0	Field length

Field Name	Type	Len	Dec	Description / Comments / Values
F40DEC	P	1	0	Number of decimals
F40LBL	A	15		Field label
F40CH1	A	20		Column heading 1
F40CH2	A	20		Column heading 2
F40CH3	A	20		Column heading 3
F40REF	A	10		Reference name
F40IAT	A	40		Input attributes list
F40OAT	A	40		Output attributes list
F40EDC	A	1		Edit code
F40EDW	A	24		Edit word
F40DFT	A	20		Default value
F40HLP	A	10		Pointer to help text
F40P#I	A	3		Partition Identifier
F40EAT	A	1		Extended field type
F40OVF	A	10		Overlay field name
F40OVP	A	3		Overlay field position
F40IDX	A	2		Index field name
F40SP1	A	20		Spare field 1
F40SP2	A	10		Spare field 2

DC@F41 - Help Text Subject Pointer

File Name DC@F41

Description Help Text Subject Pointer file.

Implemented in: 4.00 / D3

Normal Library: <<pgmlib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F41V1	Read	F41PGM,F41SEQ (unique).
DC@F41V2	Update	Same as DC@F41V1.
DC@F41V3	Read	F41PTR,F41PGM.
DC@F41V4	Update	Same as DC@F41V1.

ROS comments: Can be cleared of all data.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F41PGM	A	10		Program name
F41SEQ	P	7	0	Sequence number
F41PTR	A	10		HELP text pointer
F41SUB	A	40		Subject

DC@F42 - Export/Import Object Selection

File Name DV@F42

Description Export/Import Object Selection file.

Implemented in: 4.00/ D3

Normal Library <<dtalib>>

Logical Views

No Logical Views attached to this file.

ROS comments: Required.

Record Layout

Field	Type	Len	Dec	Description / Comments / Values
F42RUN	A	10		Run number
F42RTP	A	1		Run type
F42SOB	A	10		Source object
F42SLB	A	10		Source Library
F42TOB	A	10		Target object
F42TLB	A	10		Target Library
F42RRN	P	7	0	Data record RRN
F42IMP	A	1		Import flag
F42DTM	P	12	0	Date/time stamp
F42SQP	P	7	0	Sequence pointer

DC@F44 - Process Menu Attachments

File Name DC@F44

Description Process Menu Attachments file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F44V1	Read	F44P#I,F44MOD,F44SEQ.
DC@F44V2	Update	Same as DC@F44V1.
DC@F44V3	Read	F44P#I,F44AMO,F44AFM.
DC@F44V4	Update	Same as DC@F44V3.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F44MOD	A	10		Process name
F44SEQ	P	7	0	Sequence Number
F44AMO	A	10		Attached process name
F44AFM	A	7		Attached function
F44P#I	A	3		Partition Identifier

DC@F45 - Virtual Field RPGIII Code

File Name DC@F45

Description Virtual Field RPGIII Code file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F45V1	Read	F45P#I,F45PFL,F45PLB,F45VER, F45TYP,F45SEQ.
DC@F45V2	Update	Same as DC@F45V1.
DC@F45V3	Read	F45P#I,F45PFL,F45PLB,F45VER, F45TYP,F45DTA.
DC@F45V4	Update	Same as DC@F45V3.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F45PFL	A	10		Physical file name
F45PLB	A	10		File library
F45VER	P	7	0	Version no.
F45VTP	A	1		Version type
F45TYP	A	3		Type of source data
F45SEQ	P	6	2	Line sequence number
F45DTA	A	80		Source data
F45P#I	A	3		Partition Identifier

DC@F46 - Partition Identifier

File Name DC@F46

Description Partition Identifier file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F46V1	Read	F46P#I (unique).
DC@F46V2	Update	Same as DC@F46V1.
DC@F46V3	Read	F46SUI (unique).
DC@F46V4	Update	Same as DC@F46V3.
DC@F46V5	Read	F46MDL (unique).
DC@F46V6	Update	Same as DC@F46V5.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F46P#I	A	3		Partition Identifier
F46P#D	A	40		Partition description
F46MDL	A	10		Partition module library
F46SUI	A	1		Partition unique prefix
F46SEC	A	50		Partition security officer
F46SEL	P	7	0	Security officer length
F46DFN	A	10		Default file name
F46SAA	A	1		SAA partition
F46PEL	A	200		Panel element/area code
F46COL	A	200		SAA color for element
F46CAT	A	2000		SAA color attributes
F46MAT	A	2000		SAA mono attributes
F46IDS	A	1000		Other SAA/CUA attribute

DC@F47 - Built-In Functions Definition

File Name DC@F47

Description Built-In Functions Definition file.

Implemented in: 4.00 / D3

Normal Library: <<pgmlib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F47V1	Read	F47BIF (unique).
DC@F47V2	Update	Same as DC@F47V1.
DC@F47V3	Read	F47IDN (unique).
DC@F47V4	Update	Same as DC@F47V3.

ROS comments: Required for FFC and compiles.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F47BIF	A	20		Built-In Function name
F47IDN	P	3	0	Unique number / identifier
F47DES	A	40		Description
F47COE	A	1		Call or Exsr (C/E)
F47PGM	A	8		Program / Exsr name to use
F47SLR	A	1		Setons LR indicator (Y/N)
F47ARG	P	3	0	No. arguments (0 - 50)
F47RET	P	3	0	No. return values (0 - 50)

DC@F48 - Built-In Functions Definition

File Name DC@F48

Description Built-In Functions Definition file.

Implemented in: 4.00 / D3

Normal Library: <<pgmlib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F48V1	Read	F48BIF,F48AOR,F48SEQ (unique).
DC@F48V2	Update	Same as DC@F48V1.
DC@F48V3	Read	F48BIF,F48PNO (unique).
DC@F48V4	Update	Same as DC@F48V3.

ROS comments: Required for FFC and compiles.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F48BIF	A	20		Built-In Function name
F48AOR	A	3		Argument/return (ARG/RET)
F48SEQ	P	3	0	Sequence within ARG/RET
F48PNO	P	3	0	Sequence as a CALL parm
F48IDN	A	1		Identification number/char
F48DES	A	40		Description
F48ROO	A	1		Required/optional (R/O)
F48TYP	A	1		Alpha, Numeric or List (A/N/L)
F48MLN	P	3	0	Minimum length allowed
F48XLN	P	3	0	Maximum length allowed
F48MDP	P	1	0	Minimum decimals allowed
F48XDP	P	1	0	Maximum decimals allowed
F48PLN	P	3	0	Pass / return length
F48PDP	P	1	0	Pass / return decimals
F48DFT	A	10		Default if not specified

DC@F49 - Export Lists

File Name DC@F49

Description Export Lists file.

Implemented in: 4.00 / D3

Normal Library <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F49V1	Read	F49P#I,F49LNM (unique).
DC@F49V2	Update	Same as DC@F49V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F49P#I	A	3		Partition Identifier
F49LNM	A	10		List name
F49DSC	A	40		List description
F49TST	A	6		Target system type

DC@F50 - Export List Entries

File Name DC@F50

Description Export List Entries file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F50V1	Read	F50P#I,F50LNM,F50PSQ, F50TYP,F50PRO,F50SOB, F50SLB, F50VAR (unique).
DC@F50V2	Update	Same as DC@F50V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F50P#I	A	3		Partition Identifier
F50LNM	A	10		List name
F50SOB	A	10		Source object
F50SLB	A	10		Source Library
F50TLB	A	10		Target library
F50TYP	A	10		Object type
F50DSC	A	40		Object description
F50EFD	A	1		Export file data?
F50ECF	A	1		Export compiled form?
F50PRO	A	10		Process name
F50EXP	A	1		Object exported (Y/N)
F50VAR	A	20		System variable name
F50PSQ	S	3	0	Processing sequence
F50COU	A	1		OS/2 Check Out Usage
F50TSK	A	8		OS/2 Check out Task ID

DC@F52 - Export Substitution Variables

File Name DC@F52

Description Export Substitution Variables file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F52V1	Read	F52P#I,F52LNM,F52SVN (unique).
DC@F52V2	Update	Same as DC@F52V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F52P#I	A	3		Partition Identifier
F52LNM	A	10		List name
F52SVN	A	10		Substitution variable
F52SVL	A	50		Substitution value
F52PIM	A	1		Prompt user on import?
F52PTX	A	70		Prompt text

DC@F53 - Export CPF Commands to Execute on Import

File Name DC@F53

Description Export CPF Commands to Execute on Import file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F53V1	Read	F53P#I,F53LNM,F53EBA, F53SEQ (unique).
DC@F53V2	Update	Same as DC@F53V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F53P#I	A	3		Partition Identifier
F53LNM	A	10		List name
F53EBA	A	1		Execute before/after import?
F53SEQ	P	7	0	Execution sequence no.
F53CMD	A	400		CPF command
F53IGN	A	1		Ignore error for cmd?

DC@F55 - Application Templates Definition

File Name DC@F55

Description Application Templates Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F55V1	Read	F55ATI (unique).
DC@F55V2	Update	Same as DC@F55V1.
DC@F55V3	Read	F55DSQ.
DC@F55V4	Update	Same as DC@F55V3.

ROS comments: Can be cleared.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F55ATI	A	10		Template Identifier
F55ATD	A	40		Template description
F55DSQ	P	7	0	Display sequence
F55AEU	A	1		Accessible to end users
F55ED1	A	40		Extended description 1
F55ED2	A	40		Extended description 2
F55ED3	A	40		Extended description 3

DC@F56 - Application Templates Commands

File Name DC@F56

Description Application Templates Command file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F56V1	Read	F56ATI,F56SEQ (unique).
DC@F56V2	Update	Same as DC@F56V1.
DC@F56V3	Read	F56ATI,F56TAG.
DC@F56V4	Update	Same as DC@F56V3.

ROS comments: Can be cleared.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F56ATI	A	10		Template Identifier
F56SEQ	P	7	0	Command sequence
F56CMD	A	10		Command
F56TAG	A	3		Command tag/label
F56CMT	A	1		Command type (4 or T)

DC@F57 - Application Templates Parameter

File Name DC@F57

Description Application Templates Parameter file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F57V1	Read	F57ATI,F57SEQ,F57LIN (unique).
DC@F57V2	Update	Same as DC@F57V1.

ROS comments: Can be cleared.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F57ATI	A	10		Template Identifier
F57SEQ	P	7	0	Parameter sequence
F57LIN	P	5	0	Line number
F57PRM	A	55		Parameter details

DC@F58 - Application Templates Help Panel

File Name DC@F58

Description Application Templates Help Panel file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F58V1	Read	F58ATI,F58HPI (unique).
DC@F58V2	Update	Same as DC@F58V1.
DC@F58V3	Read	F58ATI,F58DSQ.
DC@F58V4	Update	Same as DC@F58V3.

ROS comments: Can be cleared.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F58ATI	A	10		Template Identifier
F58HPI	A	10		Help panel identifier
F58HPD	A	30		Help panel description
F58DSQ	P	7	0	Display sequence
F58HLP	A	1600		Help panel (20 X 80)

DC@F59 - Application Templates Work File

File Name DC@F59

Description Application Templates Work file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F59V1	Read	F59P#I,F59MOD,F59FMT,F59ATI, F59VAR (unique).
DC@F59V2	Update	Same as DC@F59V1.

ROS comments: Can be cleared.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F59P#I	A	3		Partition Identifier
F59MOD	A	10		Process name
F59FMT	A	7		Function name
F59ATI	A	10		Template name
F59VAR	A	10		@@ variable name
F59UVA	A	74		User value alpha
F59UVN	P	15	5	User value numeric
F59SVA	A	74		Subs. value alpha
F59SVN	P	15	5	Subs. value numeric

DC@F60 - Multilingual Extension of DC@F46

File Name DC@F60

Description Multilingual Extension of DC@F46 file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F60V1	Read	F60P#I,F60PLN (unique).
DC@F60V2	Update	Same as DC@F60V1.

ROS comments: Required in multilingual.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F60P#I	A	3		Partition identifier
F60PLN	A	4		Partition language code
F60PLD	A	20		Partition language desc
F60DBC	A	1		DBCS support required
F60RLT	A	1		RLTB support required
F60P#D	A	40		Partition description
F60IDS	A	1000		Other multilingual attributes
F60MSG	A	100		Message files
F60CHR	A	20		Character set/code page

*DC@F61 - Storage of *MTXT Multilingual Text*

File Name DC@F61

Description Storage of *MTXT Multilingual Text file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F61V1	Read	F61P#I,F61PLN,F61MTX (unique).
DC@F61V2	Update	Same as DC@F61V1.
DC@F61V3	Read	F61P#I,F61MTX,F61PLN (unique).
DC@F61V4	Update	Same as DC@F61V3.

ROS comments: Required in multilingual.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F61P#I	A	3		Partition identifier
F61PLN	A	4		Partition language code
F61MTX	A	20		*MTXT variable name
F61VAL	A	78		*MTXT value
F61LEN	P	3	0	*MTXT variable length
F61OPT	A	1		Right/left/center opt.
F61CLN	P	3	0	Length to center in

DC@F62 - Multilingual Extension of DC@F03

File Name DC@F62

Description Multilingual Extension of DC@F03 file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F62V1	Read	F62P#I,F62PLN,F62FLD (unique).
DC@F62V2	Update	Same as DC@F62V1.
DC@F62V3	Read	F62P#I,F62FLD,F62PLN (unique).
DC@F62V4	Update	Same as DC@F62V3.

ROS comments: Required in multilingual.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F62P#I	A	3		Partition identifier
F62PLN	A	4		Partition language code
F62FLD	A	10		Field name
F62DES	A	40		Field description
F62LBL	A	15		Field label
F62CH1	A	20		Column heading 1
F62CH2	A	20		Column heading 2
F62CH3	A	20		Column heading 3

DC@F63 - Multilingual Extension of DC@F04

File Name DC@F63

Description Multilingual Extension of DC@F04 file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F63V1	Read	F63P#I,F63PLN,F63HLP, F63SEQ (unique).
DC@F63V2	Update	Same as DC@F63V1.
DC@F63V3	Read	F63P#I,F63HLP,F63PLN, F63SEQ (unique).
DC@F63V4	Update	Same as DC@F63V3.

ROS comments: Required in multilingual.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F63P#I	A	3		Partition identifier
F63PLN	A	4		Partition language code
F63HLP	A	10		Help text pointer
F63SEQ	P	7	0	Sequence number
F63TYP	A	1		Text type
F63ATR	A	1		Attribute
F63TXT	A	77		Help text

DC@F64 - Multilingual Extension of DC@F27

File Name DC@F64

Description Multilingual Extension of DC@F27 file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F64V1	Read	F64P#I,F64PLN,F64PFL,F64PLB F64VER (unique).
DC@F64V2	Update	Same as DC@F64V1.
DC@F64V3	Read	F64P#I,F64PFL,F64PLB,F64VER F64PLN (unique).
DC@F64V4	Update	Same as DC@F64V3.

ROS comments: Required in multilingual.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F64P#I	A	3		Partition identifier
F64PLN	A	4		Partition language code
F64PFL	A	10		Physical file name
F64PLB	A	10		File library
F64VER	P	7	0	Version no.
F64DES	A	40		File description

DC@F65 - Multilingual Extension of DC@F15

File Name DC@F65

Description Multilingual Extension of DC@F15 file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F65V1	Read	F65P#I,F65PLN,F65PFL,F65PLB F65VER,F65LGL (unique).
DC@F65V2	Update	Same as DC@F65V1.
DC@F65V3	Read	F65P#I,F65PFL,F65PLB,F65VER F65LGL,F65PLN (unique).
DC@F65V4	Update	Same as DC@F65V3.

ROS comments: Required in multilingual.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F65P#I	A	3		Partition identifier
F65PLN	A	4		Partition language code
F65PFL	A	10		Physical file name
F65PLB	A	10		File library
F65VER	P	7	0	Version no.
F65LGL	A	10		Logical view name
F65DES	A	40		Logical view description

DC@F66 - Multilingual Extension of DC@F20

File Name DC@F66

Description Multilingual Extension of DC@F20 file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F66V1	Read	F66P#I,F66PLN,F66MOD (unique).
DC@F66V2	Update	Same as DC@F66V1.
DC@F66V3	Read	F66P#I,F66MOD,F66PLN (unique).
DC@F66V4	Update	Same as DC@F66V3.

ROS comments: Required in multilingual.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F66P#I	A	3		Partition identifier
F66PLN	A	4		Partition language code
F66MOD	A	10		Process name
F66DES	A	40		Process description

DC@F67 - Multilingual Extension of DC@F23

File Name DC@F67

Description Multilingual Extension of DC@F23 file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F67V1	Read	F67P#I,F67PLN,F67MOD F67FMT (unique).
DC@F67V2	Update	Same as DC@F67V1.
DC@F67V3	Read	F67P#I,F67MOD,F67FMT, F67PLN (unique).
DC@F67V4	Update	Same as DC@F67V3.

ROS comments: Required in multilingual.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F67P#I	A	3		Partition identifier
F67PLN	A	4		Partition language code
F67MOD	A	10		Process name
F67FMT	A	7		Function name
F67DES	A	40		Function description

DC@F68 - Multilingual Extension of DC@F31

File Name DC@F68

Description Multilingual Extension of DC@F31 file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F68V1	Read	F68P#I,F68PLN,F68MOD, F68SEQ (unique).
DC@F68V2	Update	Same as DC@F68V1.
DC@F68V3	Read	F68P#I,F68MOD,F68SEQ, F68PLN (unique).
DC@F68V4	Update	Same as DC@F68V3.

ROS comments: Required in multilingual.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F68P#I	A	3		Partition identifier
F68PLN	A	4		Partition language code
F68MOD	A	10		Process name
F68SEQ	P	7	0	Sequence Number
F68DES	A	40		Menu description

DC@F69 - Action Bar Definition

File Name DC@F69

Description Action Bar Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F69V1	Read	F69P#I,F69MOD,F69BNO, F69PNO (unique).
DC@F69V2	Update	Same as DC@F69V1.

ROS comments:

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F69P#I	A	3		Partition Identifier
F69MOD	A	10		Process name
F69BNO	P	3	0	Action Bar Number
F69PNO	P	3	0	Pull Down Number
F69BDS	A	10		Action Bar Description
F69BCD	A	3		Action Bar Code
F69PDS	A	20		Pull Down Description
F69PCD	A	3		Pull Down Code
F69TYP	A	3		Pull Entry Type
F69TK1	A	10		Pull Down Process
F69TK2	A	10		Pull Down Function
F69AFK	A	2		Accelerator Key
F69IAV	A	1		Initial Availability

DC@F70 - Compiler Work File

File Name DC@F70

Description Contains work members used during a compile

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

No Logical Views attached to this file.

ROS comments: Required to compile any object.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
SRCSEQ	S	6	2	Source sequence
SRCDAT	S	6	0	Source date
SRCDTA	A	80		Source data

DC@F71 - Action Bar Multilingual Definition

File Name DC@F71

Description Action Bar Multilingual Definition file.

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F71V1	Read	F71P#I,F71MOD,F71BNO,F71PNO, F71PLN (unique).
DC@F71V2	Update	Same as DC@F71V1.
DC@F71V3	Read	F71P#I,F71PLN,F71MOD,F71BNO, F71PNO (unique).
DC@F71V4	Update	Same as DC@F71V3.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F71P#I	A	3		Partition Identifier
F71MOD	A	10		Process name
F71BNO	P	3	0	Action Bar Number
F71PNO	P	3	0	Pull Down Number
F71PLN	A	4		Partition language code
F71BDS	A	10		Action Bar Description
F71PDS	A	20		Pull Down Description

DC@F72 - Battery License File

File Name DC@F72

Description Battery License file

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F72V1	Read	F72REC.
DC@F72V2	Update	Same as DC@F72V1.

ROS comments: Required if battery licenses are used.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F72REC	A	3		Record type
F72LEN	P	7	0	Encrypted data length
F72DTA	A	200		Encrypted data

DC@F73 - TTS Export/Import File for External Systems

File Name DC@F73

Description TTS Export/Import file for external systems

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F73V1	Read	F50P#I, F50LNM, F50PSQ, F50TYP, F50PRO, F50SOB, F50SLB, F50VAR
DC@F73V2	Update	Same as DC@F73V1.
DC@F73V3	Read	F50LNM.
DC@F73V4	Update	Same as DC@F73V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F50P#I	A	3		Partition Identifier
F50LNM	A	10		List name
F50SOB	A	10		Source object
F50SLB	A	10		Source library
F50TLB	A	10		Target library
F50TYP	A	10		Object type
F50DSC	A	40		Object description
F50EFD	A	1		Export data
F50ECF	A	1		Export compiled
F50PRO	A	10		Process name
F50EXP	A	1		Object exported
F50VAR	A	20		System variable name
F50PSQ	S	3	0	Export/Import sequence

DC@F74 - TTS Object Register

File Name DC@F74

Description TTS Object Register

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F74V1	Read	F74P#I, F74OBJ, F74TYP, F74EXT
DC@F74V2	Update	Same as DC@F74V1.
DC@F74V3	Read	F74TSK, F74P#I, F74OBJ, F74TYP, F74EXT.
DC@F74V4	Update	Same as DC@F74V1.
DC@F74V5	Read	F74TSK, F74TYP, F74P#I, F74OBJ, F74EXT.
DC@F74V6	Update	Same as DC@F74V5.

ROS comments: Can be cleared.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F74P#I	A	3		Partition Identifier
F74OBJ	A	10		Object name
F74TYP	A	2		Object type
F74EXT	A	10		Object name extension
F74OBD	A	50		Object description
F74TSK	A	8		Task identifier
F74CRS	S	12	0	Date/Time record Created
F74CRU	A	10		User who created
F74UPS	S	12	0	Date/Time record updated
F74UPU	A	10		User who updated record
F74USD	A	10		Host/User System data

DC@F75 - TTS Task Definition Log

File Name DC@F75

Description TTS Task Definition Log

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F75V1	Read	F75TSK.
DC@F75V2	Update	Same as DC@F75V1.

ROS comments: Can be cleared.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F75TSK	A	8		Task ID
F75TSD	A	50		Task description
F75USR	A	100		List of 10 authorized user/groups
F75STS	A	3		Task status
F75CRS	S	12	0	Date/Time Record Created
F75CRU	A	10		User who created
F75UPS	S	12	0	Date/Time record updated
F75UPU	A	10		User who updated record
F75USD	A	10		Host/User System data

DC@F76 - TTS Object Event Log

File Name DC@F76

Description TTS Object Event Log

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F76V1	Read	F76P#I, F76OBJ, F76EXT, F76TYP.
DC@F76V2	Update	Same as DC@F76V1.
DC@F76V3	Read	F76TSK, F76P#I, F76OBJ, F75EXT,, F76TYP, F76CRS.
DC@F76V4	Update	Same as DC@F76V1.

ROS comments: Can be cleared.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F76P#I	A	3		Partition ID
F76OBJ	A	10		Object name
F76EXT	A	10		Object name extension
F76TYP	A	2		Object type
F76TSK	A	8		Task ID
F76ECD	A	3		Event code
F76MID	A	7		Message identifier issued
F76MFN	A	10		Message file name
F76MFL	A	10		Message file library
F76MDV	A	132		Message data field values
F76CRS	S	12	0	Date/Time record created
F76CRU	A	10		User who created record
F76USD	A	20		Host/User System data

DC@F77 - TTS Export/Import Log for External Systems

File Name DC@F77

Description TTS Export/Import Log for External Systems

Implemented in: 4.00 / D3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F77V1	Read	F77P#I, F77LNM, F77OBJ, F77EXT, F77TYP.
DC@F77V2	Update	Same as DC@F77V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F77P#I	A	3		Partition ID
F77LNM	A	10		List name
F77OBJ	A	10		Object name
F77EXT	A	10		Object name extension
F77TYP	A	2		Object type
F77PRO	A	10		Process name
F77VAR	A	20		System variable name
F77MST	A	1		Message type
F77EIF	A	10		Message for Export or Import ?
F77TSK	A	8		Task ID
F77USD	A	10		Host/User System data

DC@F78 - Multilingual Textual Data for Development

File Name DC@F78

Description Multilingual Textual Data for Development

Implemented in: 4.00 / D4

Normal Library: <<pgmlib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F78V1	Read	F78TID, F78PLN
DC@F78V2	Update	Same as DC@F78V1.
DC@F78V3	Read	F78PLN, F78TID
DC@F78V4	Update	Same as DC@F78V3.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F78TID	S	4	0	Text ID
F78PLN	A	4		Partition Language Code
F78DTA	A	78		Textual Data

DC@F79 - Multilingual Textual Data Extension

File Name DC@F79

Description Multilingual Textual Data Extension

Implemented in: 4.00 / D3

Normal Library: <<pgmlib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F79V1	Read	F79TID
DC@F79V2	Update	Same as DC@F79V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F79TID	S	4	0	Text ID
F79LEN	S	2	0	Maximum data length
F79CM1	A	78		Comments
F79CM2	A	78		Comments
F79CM3	A	78		Comments
F79CM4	A	78		Comments
F79CM5	A	78		Comments

DC@F80 - Saved Working Lists

File Name DC@F80

Description Saved working lists

Implemented in: 4.00 / E3

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F80V1	Read	F80P#I, F80LST, F80SEQ
DC@F80V2	Update	Same as DC@F80V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F80P#I	A	3		Partition
F80LST	A	10		List name
F80LTP	A	1		List type
F80DYS	P	2	0	Retention period
F80BSZ	P	15	0	Block size
F80DTA	A	999		List data
F80BUD	P	7	0	Bytes used
F80SEQ	P	7	0	Sequence
F80DTE	A	5		Date written (Julian)

DC@F81 - Developer Message Header File

File Name DC@F81

Description Developer Message Header file.

Implemented in: 4.00 / F2

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F81V1	Read	F81MSN, F81MGN, F81PGL (unique)
DC@F81V2	Update	Same as DC@F81V1.
DC@F81V3	Read	F81TYP, F81OBJ, F81EXT, F81CDT(Descend), and F81CTM(Descend).
DC@F81V4	Update	Same as DC@F81V3.
DC@F81V5	Read	F81FUR, F81CDT(Descend), F81CTM(Descend) F81TYP, F81OBJ and F81EXT.
DC@F81V6	Update	Same as DC@F81V5.
DC@F81V7	Read	F81TUR, F81CDT(Descend), F81CTM(Descend) F81TYP, F81OBJ and F81EXT.
DC@F81V8	Update	Same as DC@F81V7.
DC@F81V9	Read	F81CAT, F81TUR, F81TYP, F81OBJ, F81EXT, F81CDT(Descend) and F81CTM(Descend)
DC@F81VA	Update	Same as DC@F81V9.
DC@F81VB	Read	F81CDT(Descend),

Name	Use	Keys/Description/Comments
		F81CTM(Descend)
DC@F81VC	Update	Same as DC@F81VB.

ROS Comments: Not required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F81MSN	A	8		Machine Serial number
F81MGN	S	7	0	Message number
F81PGL	A	10		Program library name
F81CDT	S	8	0	Date Created/Sent (CCYY/MM/DD)
F81CTM	S	6	0	Time Created/Sent (HH:MM:SS)
F81CJN	A	10		Job Name Created by/Sent from
F81CJ#	A	6		Job Number Created by/Sent from
F81FUR	A	10		User Created by/Sent from
F81TUR	A	10		User Message intended for
F81TYP	A	2		Object Type
F81OBJ	A	10		Object Name
F81EXT	A	10		Object Name Extension
F81SBJ	A	20		Subject of Message
F81CAT	A	3		Category
F81POT	A	1		Permanent or Temporary
F81UDT	S	8	0	Date Updated/Received (CCYY/MM/DD)
F81UTM	S	6	0	Time Updated/Received (HH:MM:SS)
F81UJN	A	10		Job Name Updated/Received by
F81UJ#	A	6		Job Number Updated/Received by
F81UUR	A	10		User Updated by/Received by

DC@F82 - Developer Message Detail File

File Name DC@F82

Description Developer Message Detail file.

Implemented in: 4.00 / F2

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F82V1	Read	F82MSN, F82MGN, F82PGL
DC@F82V2	Update	Same as DC@F82V1.

ROS Comments: Not required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F82MSN	A	8		Machine Serial number
F82MGN	S	7	0	Message number
F82PGL	A	10		Program library name
F82TXT	A	70		Message Text

DC@F83 - Developer Message Header LOG File

File Name DC@F83

Description Developer Message Header LOG file.

Implemented in: 4.00 / F2

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F83V1	Read	F83MSN, F83MGN, F83PGL, F83ADT, F83ATM, F83ACT, F83AJN, F83AJ# and F83AUR.
DC@F83V2	Update	Same as DC@F83V1.
DC@F83V3	Read	F83ADT, F83ATM.
DC@F83V4	Update	Same as DC@F83V3.

ROS Comments: Not required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F83MSN	A	8		Machine Serial number
F83MGN	S	7	0	Message number
F83PGL	A	10		Program library name
F83ADT	S	8	0	Date of Action (CCYY/MM/DD)

Field Name	Type	Len	Dec	Description / Comments / Values
F83ATM	S	6	0	Time of Action (HH:MM:SS)
F83ACT	A	3		Action (CRT, CHG or DLT)
F83AJN	A	10		Job Name Actioned by
F83AJ#	A	6		Job Number Actioned by
F83AUR	A	10		Action by User
F83CDT	S	8	0	Date Created/Sent (CCYY/MM/DD)
F83CTM	S	6	0	Time Created/Sent (HH:MM:SS)
F83CJN	A	10		Job Name Created by/Sent from
F83CJ#	A	6		Job Number Created by/Sent from
F83FUR	A	10		User Created by/Sent from
F83TUR	A	10		User Message intended for
F83TYP	A	2		Object Type
F83OBJ	A	10		Object Name
F83EXT	A	10		Object Name Extension
F83SBJ	A	20		Subject of Message
F83CAT	A	3		Category
F83POT	A	1		Permanent or Temporary
F83UDT	S	8	0	Date Updated/Received (CCYY/MM/DD)
F83UTM	S	6	0	Time Updated/Received (HH:MM:SS)
F83UJN	A	10		Job Name Updated/Received by
F83UJ#	A	6		Job Number Updated/Received by
F83UUR	A	10		User Updated by/Received by

DC@F84 - Developer Message Detail LOG File

File Name DC@F84

Description Developer Message Detail LOG file.

Implemented in: 4.00 / F2

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F84V1	Read	F84MSN, F84MGN, F84PGL, F84ADT, F84ATM, F84ACT, F84AJN, F84AJ# and F84AUR.
DC@F84V2	Update	Same as DC@F84V1.

ROS Comments: Not required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F84MSN	A	8		Machine Serial number
F84MGN	S	7	0	Message number
F84PGL	A	10		Program library name
F84ADT	S	8	0	Date of Action (CCYY/MM/DD)
F84ATM	S	6	0	Time of Action (HH:MM:SS)
F84ACT	A	3		Action (CRT, CHG or DLT)
F84AJN	A	10		Job Name Actioned by
F84AJ#	A	6		Job Number Actioned by
F84AUR	A	10		Action by User
F84TXT	A	70		Message Text

DC@F85 - Miscellaneous Function Details

File Name DC@F85

Description Miscellaneous Function Details

Implemented in: 4.00 / F2

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F85V1	Read	F85P#I, F85MOD, F85FMT, F85KV1, F85KV2
DC@F85V2	Update	Same as DC@F85V1.
DC@F85V3	Read	F85P#I, F85KV2
DC@F85V4	Update	Same as DC@F85V3.
DC@F85V5	Read	F85P#I, F85MOD, F85KV1, F85KV2, F85FMT,
DC@F85V6	Update	Same as DC@F85V5.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F85P#I	A	3		Partition
F85MOD	A	10		Process name
F85FMT	A	7		Function name
F85KV1	A	10		Key value 1
F85KV2	A	10		Key value 2
F85DTA	A	132		Data value

DC@F87 - LANSA/2 Host Connection Definition

File Name DC@F87

Description LANSA/2 Host Connection Definition

Implemented in: 5.00 / F9

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F87V1	Read	F87PCN.
DC@F87V2	Update	F87PCN.
DC@F87V3	Read	F87IN.
DC@F87V4	Update	F87IN.
DC@F87V5	Read	F87OUT.
DC@F87V6	Update	F87OUT.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F87PCN	A	10		PC name
F87DES	A	30		PC description
F87APN	A	150		Allowable partitions
F87IN	A	10		Data queue inbound
F87OUT	A	10		Data queue outbound
F87JDM	A	21		Monitor job description
F87JQM	A	21		Monitor job queue
F87OQM	A	21		Monitor output queue
F87TXI	A	21		Translation table IN
F87TXO	A	21		Translation table OUT
F87JBD	A	21		Job description
F87JBQ	A	21		Job queue
F87OTQ	A	21		Output queue

DC@F88 - Trigger Definitions

File Name DC@F88

Description Trigger Definitions

Implemented in: 5.00 / F8

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F88V1	Read	F88P#I,F88PTR (unique).
DC@F88V2	Update	F88P#I,F88PTR (unique).

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F88P#I	A	3		Partition identifier
F88PTR	A	10		Pointer rule/trigger
F88FUN	A	7		Trigger function name
F88TYP	A	30		Trigger positions array
F88TEB	A	10		Trigger enabled before array
F88TEA	A	10		Trigger enabled after array
F88NOC	S	7	0	Number of trigger conditions
F88AOR	A	60		Array of AND/OR value
F88FL1	A	200		Array of field names
F88OPN	A	60		Array of operations
F88VL1	A	200		Array of values part 1 for 01 to 10
F88VL2	A	200		Array of values part 2 for 11 to 20

DC@F89 - LANSA/2 Host Connection Services Definition Table

File Name DC@F89

Description LANSA/2 Host Connections Services Definition

Implemented in: 5.00 / F9

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F89V1	Read	F89SYS,F89OSU.
DC@F89V2	Update	Same as DC@F89V1.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F89SYS	A	20		Database system name
F89OSU	A	10		OS/2 user identifier
F89ASU	A	20		AS/400 user profile identifier
F89ASG	A	20		AS/400 group profile identifier
F89ASP	A	20		

DC@F90 - Document Definition Header

File Name DC@F90

Description Document Definition Header

Implemented in: 5.00 / F5

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F90V1	Read	F90P#I, F90DNM
DC@F90V2	Update	Same as DC@F90V1.

ROS comments: Can be cleared (DC@F91 must be cleared at same time).

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F90DNM	A	10		Document Name
F90DDC	A	40		Document Description
F90COV	A	1		Cover Page Flag
F90C01	A	78		Cover Page Line 1
F90C02	A	78		Cover Page Line 2
F90C03	A	78		Cover Page Line 3
F90C04	A	78		Cover Page Line 4
F90C05	A	78		Cover Page Line 5
F90C06	A	78		Cover Page Line 6
F90C07	A	78		Cover Page Line 7
F90C08	A	78		Cover Page Line 8
F90C09	A	78		Cover Page Line 9
F90C10	A	78		Cover Page Line 10
F90C11	A	78		Cover Page Line 11
F90C12	A	78		Cover Page Line 12
F90C13	A	78		Cover Page Line 13
F90C14	A	78		Cover Page Line 14
F90PH1	A	78		Page Heading Line 1
F90PH2	A	78		Page Heading Line 2
F90PH3	A	78		Page Heading Line 3
F90PF1	A	78		Page Footing Line 1
F90PF2	A	78		Page Footing Line 2

Field Name	Type	Len	Dec	Description / Comments / Values
F90PF3	A	78		Page Footing Line 3
F90POR	A	78		Portrait Mode Instruction
F90LND	A	78		Landscape Mode Instruction
F90PFL	A	10		Printer File Name
F90APP	A	4		Appendices Required
F90RVW	A	1		TOC Reviewed Flag
F90SPL	A	3		Spool Document
F90LIN	P	3	0	Lines per Page
F90WID	P	3	0	Width of Page
F90LPI	P	1	0	Lines per inch
F90CPI	A	4		Characters per inch
F90OVF	P	3	0	Overflow line number
F90FTP	A	10		Form type
F90CPY	P	3	0	No. of copies
F90HLD	A	3		Hold spooled file
F90PCD	A	12		PC Document Name
F90FLR	A	63		Document Folder Name
F90P#I	A	3		Partition Identifier

DC@F91 - Document Table of Contents

File Name DC@F91

Description Document Table of Contents

Implemented in: 5.00 / F5

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@F91V1	Read	F91P#I, F91DNM, F91OTY, F91OBJ, F91EXT, F91TYP, F91DSC
DC@F91V2	Update	Same as DC@F91V1.
DC@F91V3	Read	F91P#I, F91DNM, F91ORD, F91SEQ
DC@F91V4	Update	Same as DC@F91V3.
DC@F91V5	Read	F91P#I, F91DNM, F91TYP
DC@F91V6	Update	Same as DC@F91V5.
DC@F91V7	Read	F91P#I, F91OBJ, F91EXT, F91OTY
DC@F91V8	Update	Same as DC@F91V7.

ROS comments: Can be cleared (DC@F92 must be cleared at same time).

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F91DNM	A	10		Document Name
F91TYP	A	2		Information Type Code
F91OBJ	A	10		Object Name
F91EXT	A	10		Object name extension
F91OTY	A	2		Object Type
F91DSC	A	64		Object Description
F91AFL	A	1		Include all files
F91AFD	A	1		Include all fields
F91RFL	P	5	0	No. using file/field
F91ORD	P	5	0	Ordering Number
F91SEQ	P	5	0	Order sequence number
F91P#I	A	3		Partition Identifier

DC@F92 - Temporary Work File for Documents

File Name	DC@F92
Description	Temporary Work file for Documents
Implemented in:	5.00 / F5
Normal Library:	<<dtalib>>

Logical Views

None.

ROS comments: Only required for generation of documents.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F92DNM	A	10		Document Name
F92ORD	P	5	0	Ordering Number
F92SEQ	P	5	0	Order sequence number
F92TYP	A	2		Information Type Code
F92OBJ	A	10		Object Name
F92EXT	A	10		Object name extension
F92OTY	A	2		Object Type
F92DTA	A	200		Line data
F92PAG	A	13		Page reference
F92LIN	P	7	0	Line number
F92P#I	A	3		Partition Identifier

DC@F93 - Temporary Work File for Documents (for Copy to PC Document)

File Name DC@F93

Description Temporary Work file for Documents (for copy to PC Document)

Implemented in: 5.00 / F5

Normal Library: <<dtalib>>

Logical Views

None.

ROS comments: Only required for generation of documents.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
F93DTA	A	200		Line data

DC@FLE - ILE Module/Service Program List

File Name DC@FLE

Description ILE Module/Service Program List

Implemented in: 6.00 / H2

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@FLEV1	Read	FLEOWN,FLEMOD,FLESRV
DC@FLEV2	Read	Same as DC@FLEV1.

ROS comments: Not required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
FLEOWN	A	1		Owner Code (L=Lans, O=Other)
FLEMOD	A	10		Module Name
FLESRV	A	10		Service Program

DC@FOL - Object Locks

File Name DC@FOL

Description Object Locks File.

Implemented in: 6.00 / G1

Normal Library: <<dtalib>>

Logical Views

Name	Use	Keys/Description/Comments
DC@FOLV1	Read	FOLP#I,FOLTYP,FOLID1, FOLID2,FOLID3,FOLID4. (unique)
DC@FOLV2	Update	Same as DC@FOLV1.
DC@FOLV3	Read	FOLLVL,FOLJ#L,FOLMOD,FOLFMT.
DC@FOLV4	Update	Same as DC@FOLV3.

ROS comments: Required.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
FOLP#I	A	3		Partition Identifier
FOLTYP	A	20		Object Type
FOLID1	A	10		Object Identifier 1
FOLID2	A	10		Object Identifier 2
FOLID3	A	10		Object Identifier 3
FOLID4	A	10		Object Identifier 4
FOLLVL	A	4		Lock Level
FOLMOD	A	10		Process name
FOLFMT	A	7		Function name
FOLJNL	A	10		Locked by Job Name
FOLJ#L	A	6		Locked by Job Number
FOLUSL	A	10		Locked by User
FOLTDS	S	12	0	Time/Date Stamp

DC@FRB - Retain Breakpoints and Variables

File Name DC@FRB

Description Retain breakpoints and variables

Implemented in: 6.00 / H1

Normal Library: <<dtalib>>

Logical Views

None.

ROS comments: Required - for RDML debug only.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
FRBP#I	A	3		Partition Identifier
FRBUSR	A	10		User name
FRBMOD	A	10		Process name
FRBFNC	A	7		Function name
FRBTYP	A	1		Record type
FRBSEQ	P	4	0	Sequence number
FRBFLD	A	10		Variable field name
FRBTDS	S	12	0	Time/Date Stamp

X_FUNRTR - Function Routing Table

File Name X_FUNRTR

Description Function Routing Table

Implemented in: 7.00 / H3

Normal Library: <<partition module library>>

This is an OPTIONAL file. It will exist only if Function Routing is required for the partition. It is a standard AS/400 source file with a single member called X_FUNRTR.

Logical Views

None.

ROS comments: Exist only if Function Routing enabled.

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
SEQNO	S	6	2	SEU Sequence Number
DATE	S	6	0	Revision Date
DATA	A	80		Data

Data portion of each record will contain the From and To function names in upper case, separated by a comma and with no embedded blanks.

Chapter 4. Open System Utilities

Abbreviations Used in This Section

Abbreviation	Full Meaning
<<pgmlib>>	Name of LANSA program library. Often DC@PGMLIB, but can be changed during installation. Refer to Installation & Maintenance Guide for more details.
<<dtalib>>	Name of LANSA data library. Often DC@DTALIB, but can be changed during installation. Refer to Installation & Maintenance Guide for more details.
<<osulib>>	Name of LANSA open systems utility library. Often DC@OPENLIB, but can be changed during installation.
ROS	Run only system. A LANSA system that is never used for application development or enhancement.

Warning About Using Open System Utilities

The Open System Utilities library is provided to you on the following basis ....

1. That you have purchased a permanent LANSA development Licence for at least one machine in your organization.

2. Open System Utilities material may be copied for backup and recovery purposes. However, material must not be copied for purposes of providing to any other organization.

   This prohibition applies to documentation and computer definition, and/or execution objects.

3. That you understand that you (the user of any Open System Utilities feature) are solely responsible for any/all results that it produces, regardless of whether or not the LANSA system involved belongs to the organization that employs you, or to any other organization.

4. Open System Utilities allow you to perform powerful global manipulations of a LANSA system outside many of the normal internal LANSA checks. As such, you are solely responsible for any/all results produced. You must exercise due skill, care and caution when using Open System Utilities.

5. Open System Utilities are provided on an "as is" basis. No warranty is expressed or implied in the provision of these Open System Utilities.

6. You are solely responsible for the installation, maintenance, backup and recovery of all Open System Utilities provided.

7. You are solely responsible for ensuring the compatibility of any release of the Open System Utilities that you are using and the release of LANSA that you are using the Open System Utilities against. Software is provided to check on this, but you are ultimately responsible for ensuring the release levels are compatible.

Summary of Objects Provided in the Open System Utilities Library

Library (as shipped)	DC@OPENLIB
Available machine types	AS/400 and System 38
Owner of objects (as shipped)	QOTHPRDOWN
Contents (as shipped)	<see table below>

Object	Type	Description
QOPNSRC	*SRCPF	Source statements for all objects
OS@F01	*PF	Directory of LANSA objects affected by OSU
OS@A01	*DTAARA	Supported LANSA PC Levels
OS@M01	*MSGF	Messages used by OSU
OS@P0001	*CL	High speed partition export to tape
OS@P0002	*CL	High speed partition import from tape
OS@P0003	*CL	High speed partition deletion
OS@P0004	*CL	Remove observability from all objects in a library
OS@P0005	*CL	Strip system to run and compile only
OS@P0006	*CL	Strip system to run only
OS@P0007	*CL	Rename LANSA libraries
OS@P0008	*CL	Rename partition libraries
OS@P0009	*CL	Restore a process from backup media
OS@P0010	*CL	Restore a function's definition from backup media

Object	Type	Description
OS@P8001	*CL	Lock and/or unlock a complete LANSA system
OS@P8002	*CL	Retrieve system details from DC@A01 data area
OS@P8003	*CL	Retrieve partition details from DC@F46
OS@P8004	*CL	Set library list for initial system access
OS@P8005	*CL	Set library list for partition access
OS@P8006	*CL	Copy records for one partition only
OS@P8007	*CL	Copy records for all partitions
OS@P8008	*CL	Check release and PC level compatibility
OS@P8009	*CL	Copy records for a process
OS@P8010	*CL	Build list of help pointers for a process
OS@P8011	*CL	Build list of help pointers for a function
OS@P8012	*RPG	Update a partition's libraries
OS@P8014	*RPG	Update security file partition libraries

Chapter 5. Open Systems Internal Database File Layouts

OS@F01 - Directory of LANSA Objects Affected by OSU

File Name	OS@F01
Description	Directory of LANSA Objects affected by OSU
Implemented in	4.00 / D3
Normal Library	<<osulib>>

Logical Views

Name	Use	Keys/Description/Comments
OS@F01V1	Read	O01OBJ, O01TYP.
OS@F01V2	Update	Same as DC@F01V1.

ROS comments:

Record Layout

Field Name	Type	Len	Dec	Description / Comments / Values
O01OBJ	A	10		Object name
O01TYP	A	8		Object type
O01NLB	A	1		Normal library (P=Pgm, D=Data)
O01RT1	A	1		Run and Compile system
O01RT2	A	1		Run only system
O01DEV	A	1		Development

OS@A01 - Supported LANSA PC Levels

Object Name *OS@A01*

Description Supported LANSA PC Levels

Implemented in 4.00 / D9

Normal Library <<osulib>>

ROS comments

Record Layout

Data Area	Type	Len	Dec	Description
OS@A01	A	20		Supported PC Levels

The Golden Rule

Run Backups First.

Chapter 6. Functional Descriptions of Programs Provided

High Speed Partition Export to Tape

Name	OS@P0001
Type	Control Language Program
Source provided	Yes

Files Used

All files in the <<dtalib>> are used by this job.

Functional Description

Library QTEMP is cleared.

This program receives the identifier of a partition and copies all required data from the LANSA data library to temporary files in library QTEMP.

When all data has been copied to QTEMP a merged list of the following is created. Partition module library, partition default file library, and all other specified libraries. The merged list is then stored in a data area in QTEMP as a "directory" of what should be found on the export tape. This data area is called OS@DIR.

All objects from QTEMP are saved to the tape device specified, followed by all libraries in the resulting merged list. Once this operation has completed a message is sent to the system operator asking whether a repeat of all save operations is required (IE Do you want another copy ?). This allows multiple save tapes to be easily prepared from one export run.

Warnings That Apply

- Remember THE GOLDEN RULE.
- You must back up <dtalib> before using this utility.
- <osulib> must be in the library list when running this job.

- The partition specified must exist.
- This routine does not do any security checking.
- No one should be using the partition at the time of export.
- Tapes are not initialized by the export operation.
- This program is best run in batch.
- If you use TGTRLS(*PRV) option, ensure that all objects being saved have been created with this option also.

Parameters

No	Type	Len	Dec	Description
1	Alpha	3		Identifier of partition to be exported.
2	Alpha	10		Name of tape device to be used for export.
3	Alpha	1		Indicates the TGTRLS that the save will use. The flag values and corresponding TGTRLS parameters are as follows. "Y" indicates TGTRLS(*PRV) "N" indicates TGTRLS(*CURRENT) "A" indicates TGTRLS(V1R3M0) "B" indicates TGTRLS(V2R1M0) "C" indicates TGTRLS(V2R1M1) "D" indicates TGTRLS(V2R2M0) "E" indicates TGTRLS(V2R3M0) "F" indicates TGTRLS(V3R0M5) "G" indicates TGTRLS(V3R1M0) ***WARNING:*** *LANSA does no checking as to the validity of the TGTRLS selected. If a*

No	Type	Len	Dec	Description
				TGTRLS is selected that is not supported by this AS400 fatal errors will occur.
4	Packed	15	5	Number of extra libraries to be saved.
5	Alpha	200		List of 20 x alpha(10) extra libraries. This list and the partition module library and default file library are merged to form the final list of libraries to be saved.

Examples

Export partition DEM to TAP01 and no other libraries.

CALL OS@P0001 (DEM TAP01 N 0 ' ')

Export partition DEM to TAP01 and also library QGPL.

CALL OS@P0001 (DEM TAP01 N 1 QGPL)

Export partition DEM to TAP01 and also libraries QGPL and TEST.

CALL OS@P0001 (DEM TAP01 N 2 'QGPL TEST')

High Speed Partition Import from Tape

Name	OS@P0002
Type	Control Language Program
Source provided	Yes

Files Used

All files in the <<dtalib>> are used by this job.

Functional Description

An attempt to exclusively lock the LANSA system will be made by this job. If unsuccessful, the job will terminate at this point.

All objects saved in QTEMP are restored from the specified tape device to library QTEMP.

All libraries in OS@DIR are restored. (See OS@P0001 Technical Description for details regarding OS@DIR).

This program then copies the data in QTEMP to the LANSA data library, effectively recreating the previously deleted (and saved) partition.

Warnings That Apply

- Remember THE GOLDEN RULE.
- You must backup <dtalib> before using this utility.
- <osulib> must be in the library list when running this job.
- The partition specified must not already exist in the target system.
- Existing partitions must have different identifiers to the one about to be restored.

- Partition to be imported must not have the same unique prefix as an existing partition (even if the partition identifiers are different).
- This routine does not do any security checking.
- No one should be using the system at the time of import.
- This program is best run in batch.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		Name of tape device to be used for import.

Examples

Import previously saved partition from TAP01.

```
CALL OS@P0002 (TAP01)
```

High Speed Partition Deletion

Name	OS@P0003
Type	Control Language Program
Source provided	Yes

Files Used

All files in the <<dtalib>> are used by this job.

Functional Description

An attempt to exclusively lock the LANSA system will be made by this job. If unsuccessful the job will terminate at this point.

Job will end if the partition to be deleted is the SYS partition.

All data, excluding the data for the partition specified, from the LANSA data library is copied to temporary files in library QTEMP.

All data from QTEMP is then copied back to the <<dtalib>>, effectively removing the specified partition from the LANSA system.

LANSA system will be released.

Warnings That Apply

- Remember THE GOLDEN RULE.
- You must backup <dtalib> before using this utility.
- <osulib> must be in the library list when running this job.
- Not to be used on partition SYS (this is checked).
- The partition specified must exist.
- This routine does not do any security checking.

- No one should be using the system at the time of deletion.
- This program is best run in batch.
- Program failure will leave system in a corrupted state. To recover from corrupted state restore <<dtalib>>.

Parameters

No	Type	Len	Dec	Description
1	Alpha	3		Identifier of partition to be deleted.

Examples

Delete partition DEM.

```
CALL OS@P0003 (DEM)
```

Remove Observability from All Objects in a Library

Name	OS@P0004
Type	Control Language Program
Source provided	Yes

Functional Description

All programs in the specified library will have their symbolic debug table removed. This table can account for up to 60% of a program's object size. For more information on this topic refer to the appropriate IBM manual.

Warnings That Apply

- Remember THE GOLDEN RULE.
- <osulib> must be in the library list when running this job.
- The library specified must exist.
- If you want to debug any programs in this library after this job has completed they will need to be recompiled.
- This utility can be run repeatedly with no problems, even over libraries already stripped.
- This job can be run on any library in the system.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		Name of library to be stripped.

Examples

Remove observability from objects in library TEST.

```
CALL OS@P0004 (TEST)
```

Strip System to Import, Compile and Run Only

Name OS@P0005

Type Control Language Program

Source provided Yes

Files Used

OS@F01 Directory of LANSA Objects affected by OSU

Functional Description

This program reads the file OS@F01 and according to the flags set in this file for each object, decides whether or not the object is needed for a run and compile system. If not the object is removed from the system, unless the object is a file (unused files are cleared).

As a last step in reducing disk usage OS@P0004 is run over <<pgmlib>> to reduce the resulting run time system to its minimum size.

The resulting system can only do the following. . .

- Import files, fields, processes and functions
- Compile processes, functions, and I/O modules
- Execute processes and functions

Warnings That Apply

- Remember THE GOLDEN RULE.
- Backup the entire LANSA system first.
- <osulib> must be in the library list when running this job.

- Run a LANSA REORG before using this utility.
- Once completed the system will be available for import, run and compile only. No other functionality.
- No one should be using the system at time of run.
- This routine does not do any security checking.

Parameters

No	Type	Len	Dec	Description
1	Alpha	1		Delete multilingual components (Y/N)

Examples

Strip system to run, import and compile only, but with multilingual components still available

CALL OS@P0005 (N)

Strip System to Import and Run Only

Name	OS@P0006
Type	Control Language Program
Source provided	Yes

Files Used

OS@F01 - Directory of LANSA Objects affected by OSU

Functional Description

This program reads the file OS@F01 and according to the flags set in this file for each object, decides whether or not the object is needed for a run only system. If not then object is removed from the system, unless the object is a file (unused files are cleared).

As a last step in reducing disk usage OS@P0004 is run over <<pgmlib>> to reduce the resulting run time system to its minimum size.

The resulting system can only do the following. . .

- Import files, fields, processes and functions
- Execute processes and functions

Warnings That Apply

- Remember THE GOLDEN RULE.
- Backup the entire LANSA system first.
- <osulib> must be in the library list when running this job.
- Run a LANSA REORG before using this utility.
- Once completed the system will be available for import and run only. No other functionality.

- No one should be using the system at time of run.

Parameters

No	Type	Len	Dec	Description
1	Alpha	1		Delete multilingual components (Y/N)

Examples

Strip system to run only with no multilingual components

CALL OS@P0006 (Y)

Rename LANSA Libraries

Name OS@P0007

Type Control Language Program

Source provided. Yes

Functional Description

The LANSA system is locked exclusively.

The <<pgmlib>> and <<dtalib>> are renamed to the specified names.

The system definition data area (DC@A01) is updated to reflect the new names for the <<pgmlib>> and <<dtalib>>.

All the LANSA commands are then changed to point to the correct <<pgmlib>>.

The LANSA system is released.

Warnings That Apply

- Remember THE GOLDEN RULE.
- Backup the system first
- <osulib> must be in the library list when running this job.
- The LANSA libraries must not be in the library list of any other job on the system.
- No one should be using the system.
- Any software that invokes this LANSA system may need to be changed.
- Any software that uses the old library names may need to be changed.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		New <<pgmlib>>
2	Alpha	10		New <<dtalib>>

Examples

Rename the LANSA libraries to LANPGMPROD and LANDTAPROD

CALL OS@P0007 (LANPGMPROD LANDTAPROD)

Rename Partition Libraries

Name OS@P0008

Type Control Language Program

Source provided Yes

Files Used

DC@F02	System Security
DC@F05	Validation Check Directory
DC@F12	File Definition
DC@F14	File Definition List
DC@F15	Logical File Definition
DC@F17	Foreign Key Check Definition
DC@F18	Access Route Definition
DC@F19	Batch Control Logic Definition
DC@F21	Module File Usage
DC@F27	File Version Definition
DC@F45	Virtual Field RPGIII Code File
DC@F64	Multilingual Extension of DC@F27
DC@F65	Multilingual Extension of DC@F15
DC@F74	TTS Object Register
DC@F75	TTS Object Event Log

Functional Description

This program receives two library names. The first name is the old partition data or module library, the second is the new name for this library.

Any occurrence of the old library specified in the files mentioned above are changed to the new library.

Warnings That Apply

- Remember THE GOLDEN RULE.
- <osulib> must be in the library list when running this job.
- No one should be using the system at time of run.
- This routine does not do any security checking.
- This routine does not rename the actual library, only the references to it in LANSA internal data base.
- Renames any occurrences of this library name in the LANSA internal data base.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		Old partition library
2	Alpha	10		New partition library

Examples

Rename partition library OLDPARTLIB to NEWPARTLIB

CALL OS@P0008 (OLDPARTLIB NEWPARTLIB)

Rename partition library DEVLANSA to DEVLANDATA

CALL OS@P0008 (DEVLANSA DEVLANDATA)

Rename partition library DEVLIB to PRODLIB

CALL OS@P0008 (DEVLIB PRODLIB)

Restore a Process from Backup Media

Name	OS@P0009
Type	Control Language Program
Source provided	Yes

Files Used

DC@F02	System Security
DC@F04	Help Text Storage
DC@F20	Module Definition
DC@F21	Module File Usage
DC@F23	Module Format Definition
DC@F25	Process Field Usage
DC@F29	Function Control Commands
DC@F30	Function Control Parameters
DC@F31	Process Additional Menu Options
DC@F32	Process Parameter Definition
DC@F40	Process / Function Field Definition
DC@F44	Process Attachments
DC@F63	Multilingual Extension of DC@F04
DC@F66	Multilingual Extension of DC@F20
DC@F67	Multilingual Extension of DC@F23
DC@F68	Multilingual Extension of DC@F31
DC@F69	Action Bar Definition

DC@F71 Action Bar Multilingual Definition

DC@F85 Miscellaneous Function Details

Functional Description

The above mentioned files are restored into temporary files in QTEMP from the specified backup media.

The records relating to the specified process, in the specified partition, are copied from the temporary files in library QTEMP into the corresponding LANSA data library files.

This effectively redefines the process to LANSA.

NOTE : Only the definition is restored. The process (and its associated functions) will need to be recompiled before being used.

Warnings That Apply

- Remember THE GOLDEN RULE.
- <osulib> must be in the library list when running this job.
- The partition specified must exist.
- The process specified must not exist.
- This routine does not do any security checking.
- Job best run interactively.
- Definition only - not compiled objects.

Parameters

No	Type	Len	Dec	Description
1	Alpha	4		Type of backup media (TAPE, SAVF).
2	Alpha	10		Name of device / save file name.
3	Alpha	10		Save file library.
4	Alpha	3		Name of partition where process resides.
5	Alpha	10		Name of process to be restored.

Examples

Restore process TEST01 from partition DEM from tape device TAP01:

CALL OS@P0009 (TAPE TAP01 ' ' DEM TEST01)

Restore process TEST01 from partition DEM from save file SAVEDATA in library QGPL:

CALL OS@P0009 (SAVF SAVEDATA QGPL DEM TEST01)

Restore a Function's Definition from Backup Media

Name OS@P0010

Type Control Language Program

Source provided Yes

Files Used

DC@F02	System Security
DC@F04	Help Text Storage
DC@F20	Module Definition
DC@F21	Module File Usage
DC@F23	Module Format Definition
DC@F25	Process Field Usage
DC@F29	Function Control Commands
DC@F30	Function Control Parameters
DC@F40	Process / Function Field Definition
DC@F63	Multilingual Extension of DC@F04
DC@F66	Multilingual Extension of DC@F20
DC@F67	Multilingual Extension of DC@F23
DC@F85	Miscellaneous Function Details

Functional Description

The above mentioned files are restored into temporary files in QTEMP from the specified backup media.

The records relating to the user specified function, in the specified process and partition, are copied from the temporary files in library QTEMP into the corresponding LANSA data library files.

This effectively redefines the function to LANSA.

NOTE : Only the definition is restored. The function will need to be recompiled before it can be executed.

Warnings That Apply

- Remember THE GOLDEN RULE.
- <osulib> must be in the library list when running this job.
- The partition specified must exist.
- The process specified must exist.
- The function specified must not exist.
- This routine does not do any security checking.
- Job best run interactively.
- Definition only - not compiled objects.

Parameters

No	Type	Len	Dec	Description
1	Alpha	4		Type of backup media (TAPE, SAVF).
2	Alpha	10		Name of device / save file name.
3	Alpha	10		Save file library.

No	Type	Len	Dec	Description
4	Alpha	3		Name of partition where process resides.
5	Alpha	10		Name of process to be restored.
6	Alpha	7		Name of function to be restored.

Examples

Restore function FUNC001 from process TEST01 from partition DEM from tape device TAP01

CALL OS@P0010 (TAPE TAP01 ' ' DEM TEST01 FUNC001)

Restore function FUNC002 from process TEST01 from partition DEM from save file SAVEDATA in library QGPL

CALL OS@P0010 (SAVF SAVEDATA QGPL DEM TEST01 FUNC002)

Lock and/or Unlock a Complete LANSA System

Name	OS@P8001
Type	Control Language Program
Source provided	Yes

Functional Description

This program receives a lock or unlock request. If the request is 'LOCK', an exclusive lock is placed on the data area DC@A01 in <<pgmlib>>. If the request is 'FREE', then the data area DC@A01 is de-allocated.

Warnings That Apply

- The <<pgmlib>> must be in the library list
- No one should be using the system at time of 'LOCK' request.
- Once the system is locked, no LANSA jobs will execute successfully.
- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	4		'LOCK' or 'FREE' the LANSA system

Retrieve System Details from DC@A01 Data Area

Name OS@P8002

Type Control Language Program

Source provided Yes

Functional Description

Retrieves the following information from the system definition data area.

- <<pgmlib>>
- <<dtalib>>
- Current Release
- Current PC Level

These values are returned to the calling program.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		<<pgmlib>>
2	Alpha	10		<<dtalib>>
3	Alpha	3		Current Release
4	Alpha	2		PC Level

Retrieve Partition Details from DC@F46

Name OS@P8003

Type Control Language Program

Source provided Yes

Files Used

DC@F46 Partition Identifier

Functional Description

Retrieves the passed partition's record from DC@F46 to return the partition data library and module library.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	3		Partition Identifier
2	Alpha	10		Partition data library
3	Alpha	10		Partition module library

Set Library List for Initial System Access

Name	OS@P8004
Type	Control Language Program
Source provided	Yes

Functional Description

Sets the <<pgmlib>> and the <<dtalib>> to be highest libraries in the library list.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		<<pgmlib>>
2	Alpha	10		<<dtalib>>

Set Library List for Partition Access

Name OS@P8005

Type Control Language Program

Source provided Yes

Functional Description

Includes the partition data and module libraries into the library list.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		Partition data library
2	Alpha	10		Partition module library

Copy Records for One Partition Only

Name OS@P8006

Type Control Language Program

Source provided Yes

Files Used

All files in the <<dtalib>> are used by this job.

Functional Description

Copies <<dtalib>> records to temporary files in QTEMP.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	3		Partition Identifier
2	Alpha	10		File name
3	Alpha	10		Field name
4	Alpha	10		Source library
5	Alpha	10		Target library
6	Alpha	3		Equal or not equal
7	Alpha	8		Member option
8	Alpha	4		Create file

Copy Records for All Partitions

Name OS@P8007

Type Control Language Program

Source provided Yes

Files Used

All files in the <<dtalib>> are used by this job.

Functional Description

Copies the records for all partitions to temporary files in QTEMP.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		File name
2	Alpha	10		Source library
3	Alpha	10		Target library
4	Alpha	8		Member option
5	Alpha	4		Create file

Check Release and PC Level Compatibility

Name	OS@P8008
Type	Control Language Program
Source provided	Yes

Functional Description

Compares the LANSA system release and PC level to the levels supported by the Open Systems Utilities. These levels are stored in data area OS@A01 in <<osulib>>.

If the levels do not correspond, OSU will send a message to the system operator stating that OSU and LANSA are incompatible.

The user then has the option to continue or end the job. Unless you are absolutely sure that it is safe to continue with the job even though the levels are incompatible, end the job at this point.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	3		Release Level
2	Alpha	2		PC Level

Copy Records for a Process

Name OS@P8009

Type Control Language Program

Source provided Yes

Functional Description

This routine is passed all Parameters specifying the copy criteria. That is, file name, source library, target library, where parameter (INCREL) details.

These parameters are used to build a specific copy command that is used for copying process details.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		File name
2	Alpha	10		Source Library
3	Alpha	10		Target Library
4	Alpha	256		Where parameter of copy (INCREL)

Build List of Help Pointers for a Process

Name OS@P8010

Type Control Language Program

Source provided Yes

Files Used

DC@F20 Module definition

Functional Description

This routine builds a list of up to 50 help pointers for a process. This is done by processing the file DC@F20.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		Process
2	Alpha	3		Partition ID
3	Alpha	500		List of help pointers

Build List of Help Pointers for a Function

Name OS@P8011

Type Control Language Program

Source provided Yes

Files Used

DC@F23 Module format Definition

Functional Description

This routine builds a list of up to 50 help pointers for a function. This is done by processing the file DC@F23.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		Process
2	Alpha	7		Function
3	Alpha	3		Partition ID
4	Alpha	500		List of help pointers

Update a Partition's Libraries

Name	OS@P8012
Type	RPG Program
Source provided	Yes

Functional Description

This routine reads a file sequentially. During the read it looks for a certain library name and replaces it with the new library name. The changed record is then updated.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		Old library name
2	Alpha	10		New library name
3	Numeric	15	5	Position of library in file

Update Security File Partition Libraries

Name OS@P8014

Type RPG Program

Source provided Yes

Files Used

DC@F02 System security

Functional Description

This routine reads file DC@F02 sequentially. During the read it looks for a certain library name and replaces it with a new library name. The changed record is then updated.

Warnings That Apply

- Do not call directly. This is a utility program for the OS@P00nn series programs in this OSU package.

Parameters

No	Type	Len	Dec	Description
1	Alpha	10		Old library name
2	Alpha	10		New library name

Chapter 7. Direct Calling of LANSA Functions

Overview

Disclaimer

All the information about directly invoking LANSA processes and functions has been tested and verified to the best of our ability. However, no guarantee or warranty is expressed or implied due to the many site defined variances that occur between LANSA installations.

The testing of applications invoked in this way is the responsibility of the user site.

All examples in this section of the guide are in RPG/400 or Control Language. This is not a restriction. User defined programs that use this method to execute LANSA applications can be written in any language available on the AS/400.

What Is Direct Calling?

Direct calling is the method used to execute LANSA processes and functions without using the LANSA command or program.

That is, a user defined 3GL program written in RPG, CL, COBOL, PL/1 etc. can be used to directly invoke a LANSA process or function.

Requirements

Users of these guidelines must have a sound working knowledge of the LANSA product.

These guidelines are for systems using LANSA Release 5.0 / PC Level F6 running on an AS/400 with OS/400 V1R3M0 or higher.

Processes must be compiled. It is impossible to invoke a LANSA process using this method if the AS/400 object for the process does not exist.

Any LANSA functions that are executed using this method must be created with FUNCTION OPTIONS(*DIRECT).

These procedures are not available for processes and/or functions that require parameters, other than working lists or data structures.

All steps and guidelines must be considered when using this method to execute LANSA processes or functions.

Determining the AS/400 Object Name of a Process & Function

Process AS/400 Object Names

The AS/400 object name of a process can be determined by displaying the contents of a data area with the same name as the process, that is to be directly invoked, in the partition module library.

Example:

The process to be invoked is PSLSYS. It resides in partition DEM and the partition module library is DC@DEMOLIB.

DSPDTAARA DC@DEMOLIB/PSLSYS

This will return a screen similar to the following:

```
                    Display Data Area
                                                    System : SYDASD25
Data area . . . . . :   PSLSYS
  Library . . . . . :     DC@DEMOLIB
Type  . . . . . :   *CHAR
Length  . . . . . :   100
Text  . . . . . :

          Value
Offset     *...+....1....+....2....+....3....+....4....+....5
    0     'CP@D00001YYNNNNNNN                                '
   50     '                                                  '
```

The process object name is positioned in bytes 2 thru 9. In this case the object name is "P@D00001".

LANSA allows recursive calling of processes therefore a two digit suffix (01 thru 09) is added to this process name when it is created.

Therefore in this case the processes AS/400 object name would be:

P@D0000101

If the application demanded that this particular process be called recursively, then subsequent calls would be to programs:

P@D0000102 -> P@D0000109

Note: This value only applies to this process in this partition on this machine. This value will vary between CPUs, and partitions. Also note that this name may change after exporting/importing the process between partitions, LANSA systems, or sites.

Function AS/400 Object Names

LANSA functions object names are much easier to determine than process object names.

LANSA functions created with the FUNCTION OPTIONS(*DIRECT) option have the following naming convention:

@FFFFFFF

Where:

- @ is the prefix
- *FFFFFFF* is the function name

That is, the object name is the function name with an "@" symbol as its prefix.

Example:

A function EMPLIST has been defined and created with FUNCTION OPTIONS(*DIRECT) specified. Its AS/400 object will be @EMPLIST.

Technical Considerations

Warnings

If you use this method to run LANSA applications, be very careful. Be absolutely sure of what you are doing. Do not bypass any part of these guidelines because they don't look important or seem pointless.

No security checking is performed.

No library list manipulation is performed. The library list must be correct. IE The following libraries must be included in the library list.

- LANSA Program Library
- LANSA Data Library
- Partition module library
- Partition data library (optionally)

The environment for the application is your responsibility. LANSA no longer has any control, the user defined 3GL program is in control.

In the case of multilingual partitions, the user defined setup program will also need to perform any overrides for message files.

IE. LANSA Message file DC@M01 may need to be overridden to the appropriate version. (e.g. French may use DC@M01FRA). Refer to the LANSA Multilingual Design and Developer manuals for more information of the type of processing that is necessary.

It is not recommended that Function Routing be used with this method of running LANSA applications. When LANSA is invoked normally by use of the LANSA command, the Function Routing Table is loaded during the LANSA initiation and is available till LANSA is terminated. If you use this method of calling LANSA applications, the Function Routing Table will not have been loaded for you.

Technical Information

When LANSA invokes a process or a function it passes some data structures that contain information that is vital for the function to execute as expected. Information like the following. . .

- Partition information (command key descriptions, data and module library names, SAA/CUA information etc)
- Multilingual information
- Environment information (OS/400 Version/Release)
- LANSA system information

Therefore, in order for a user defined 3GL program to successfully invoke a LANSA process or function directly this information needs to be available.

Note : There is some sample program code shipped with the LANSA Product, and shown later in this guide, that can be used to set up this required system information.

Calling Processes

A LANSA process is called with five parameters. A process would be called similar to the following:

RPG/400

```
CALL 'PRONAME'
PARM            DC@IDS
PARM            DC@EDS
PARM            PRNMIN
PARM            PR@IDS
PARM            DUMMY1
```

Control Language

```
CALL PGM(PRONAME) +
PARM(&DC@1DS &DC@EDS &PRNMIN &PR@IDS @DUMMY1)
```

where

PRONAME	Is the process object name	
DC@IDS	System information	A(1024)
DC@EDS	Extra System information	A(2500)
PRNMIN	Process name information	A(14)
PR@IDS	Process Information	A(5000)
DUMMY1	Dummy parameter	A(1)

Calling Functions

A LANSA function has three standard parameters. A function would be called similar to the following:

RPG/400

```
CALL
'FUNNAME'
PARM            DC@IDS
PARM            DC@EDS
PARM            PR@IDS
```

Control Language

```
CALL PGM(FUNNAME) +
PARM(&DC@IDS &DC@EDS &PR@IDS)
```

where

FUNNAME	Is the function object name	
DC@IDS	System information	A(1024)
DC@EDS	Extra System information	A(2500)
PR@IDS	Process Information	A(5000)

Calling Functions and Passing Data Structures / Working Lists

Data Structures

Other parameters are included when using advanced programming techniques that include passing data structures and/or working lists.

A function created that receives a data structure would be called with a parameter list similar to the one in the following example.

Example

Data structure received:	PSLMST	
Data structure elements:	EMPNO	A(5)
	SURNAME	A(20)
	GIVENAME	A(20)
	BIRTHDT	A(6)
	SALARY	S(9,2)

RPG/400

```
CALL
'FUNNAME'
PARM            DC@IDS
PARM            DC@EDS
PARM            PR@IDS
PARM            PSLMST
```

Control Language

CALL PGM(FUNNAME) +

PARM(&DC@IDS &DC@EDS &PR@IDS &PSLMST)

where

FUNNAME	Is the function object name	
DC@IDS	System information	A(1024)
DC@EDS	Extra System information	A(2500)
PR@IDS	Process Information	A(5000)
PSLMST	Passed data structure	A(???)

The 'Passed data structure' parameter is repeated for each of the passed data structures.

The size of the 'Passed data structure' parameter in this scenario could be 60 or 64 bytes. IE 60 bytes if the PSLMST data structure is an externally defined file (OTHER) or 64 bytes for LANSA files. The extra 4 bytes is the @@UPID field used by all files created using LANSA. Each of the fields in the data structure (file) contribute to the aggregate length of the 'passed data structure'.

Working Lists

A function created that receives a working list would be called with a parameter list similar to the one in the following example.

Example

Working list received:	PSLSKL	
Working list elements:	SKILCODE	A(4)
	SKILDESC	A(20)
	COMMENT	A(40)

RPG/400

```
CALL
'FUNNAME'
PARM            DC@IDS
PARM            DC@EDS
PARM            PR@IDS
PARM            PSLSKL
PARM            PSLNUM
PARM            PSLPOS
```

Control Language

```
CALL PGM(FUNNAME) +
PARM(&DC@IDS &DC@EDS &PR@IDS &PSLSKL +
&PSLNUM &PSLPOS)
```

where:

FUNNAME	Is the function object name	
DC@IDS	System information	A(1024)
DC@EDS	Extra System information	A(2500)
PR@IDS	Process Information	A(5000)
PSLSKL	Passed working list	n * A(???)
PSLNUM	Number of entries in list	P(7,0)
PSLPOS	Position in list	P(7,0)

The 'working list' parameters are repeated for each passed working list.

The PSLSKL parameter is defined as the aggregate length of all fields in the working list.

The size of the PSLSKL parameter of this field will be 64 bytes. In RPG terms this field is a multiple occurrence data structure where each of the fields are sub-fields of PSLSKL.

*The 'n occurrences' is the number of entries in the working list that have been passed.

Data Structures and Working Lists

A function created that receives both working lists and data structures would be called with a parameter list similar to the one in this example.

Example

Working list received:	PSLSKL	
Working list elements:	SKILCODE	A(4)
	SKILDESC	A(20)
	COMMENT	A(40)
Data structure received:	PSLMST	
Data structure elements:	EMPNO	A(5)
	SURNAME	A(20)
	GIVENAME	A(20)
	BIRTHDT	A(6)
	SALARY	S(9,2)

RPG/400

```
CALL
'FUNNAME'
PARM            DC@IDS
PARM            DC@EDS
PARM            PR@IDS
PARM            PSLSKL
PARM            PSLNUM
PARM            PSLPOS
PARM            PSLMST
```

Control Language

```
CALL PGM(PRONAME) +
PARM(&DC@IDS &DC@EDS &PR@IDS &PSLSKL +
&PSLNUM &PSLPOS &PSLMST)
```

where:

FUNNAME	Is the function object name	
DC@IDS	System information	A(1024)
DC@EDS	Extra System information	A(2500)
PR@IDS	Process Information	A(5000)
PSLSKL	Passed working list	n * A(???)
PSLNUM	Number of entries in list	P(7,0)
PSLPOS	Position in list	P(7,0)
PSLMST	Passed data structure	A(???)

A breakdown of the standard parameters is supplied in another section of this guide.

Exchange Lists

A thorough explanation of exchange lists is given in the LANSA Technical Guide under the EXCHANGE command.

When using exchange lists it is important to remember that extra calls to M@EXCHL are needed when directly calling a function.

The difference between using LANSA to RUN a function and directly calling the function, is that LANSA controls the transfer of the external (3GL) exchange list to the LANSA exchange list and vice versa.

When a function is called directly these transfers must be done through additional calls to the M@EXCHL program.

One call to M@EXCHL is done just before the call to the 3GL program to map the external exchange list into the LANSA exchange list. Another is done right after the call to the 3GL program to map the LANSA exchange list back into the external exchange list.

The parameters required for these M@EXCHL calls are ...

Parm No	Type	Min Len	Max Len	Comments
1	Alpha	3	3	'$AC' : Transfer onto exchange list '$RC' : Receive from exchange list
2	Alpha	1	1024	DC@IDS - System Information
3	Alpha	1	2500	DC@EDS - Extra System Information
4	Alpha	1	5000	PR@IDS - Process Information

The correct sequence of calls when using exchange lists within a 3GL program to pass field values to a LANSA function is:

1. CALL M@EXCHL 'CLR'
2. CALL M@EXCHL 'PUT'
3. CALL M@EXCHL '$AC'
4. CALL Function (or Process)
5. CALL M@EXCHL '$RC'
6. CALL M@EXCHL 'GET' ...

Setting up the Required System Information

Shipped with the LANSA product are two source members, UD@CALL1 and UD@CALL2. The source is available in file DC@F28 in the LANSA Data Library. (A listing is included in a later section of this guide). The objects are not shipped.

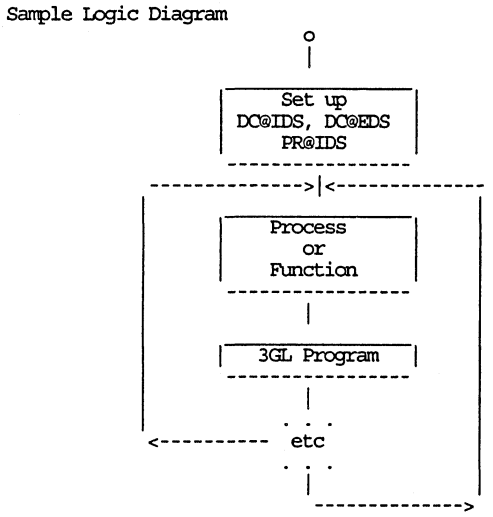
The shipped programs can be used to set up the DC@IDS and DC@EDS information. However, the user defined 3GL program, that calls the LANSA process or function, will have to supply the PR@IDS - Process Information. (A sample set up for PR@IDS is supplied later in this guide.)

To automatically load the DC@IDS and DC@EDS system information include a call to programs UD@CALL1 and UD@CALL2 respectively in the user defined 3GL program that will invoke the LANSA process or function.

Hints

The user defined 3GL program should only load the required information for this type of processing once within a job. Once the system information is loaded it can be passed between all programs within the job.

The following diagram shows a possible logic flow of an application that uses user defined 3GL programs to execute a LANSA application.



Data Structures (DC@IDS, DC@EDS & PR@IDS)

DC@IDS & DC@EDS - System Information

The /COPY source member DC@ISPEC can be found in file DC@F28 in the LANSA Data Library.

```
I*=======================================================
I* LANSA INFORMATION DATA STRUCTURE PASSED TO EVERY PROGRAM
I*=======================================================
I*
I*     DC@IDS - SYSTEM INFORMATION
I*     ---------------------------
I*
I*  Type  Len  Dec   Field  Description
I*  ~~~~  ~~~  ~~~   ~~~~~  ~~~~~~~~~~~
I*    A    10        DC@PGL  Name of program library
I*    A    10        DC@DTL  Name of data library
I*    A    10        DC@MDL  Name of module library
I*    A    10        DC@OWN  Name of system owner profile
I*    A     5        DC@PRO  Name of product
I*    S     3   1    DC@LVL  Release level of product
I*    A    30        DC@ORG  Name of company / organization
I*    A    10        DC@SEC  System security officer
I*    A    10        DC@LIC  CPU Licence number
I*
I*    A     2        DC@CEX  Exit command key
I*    S     2   0    @E      Exit command key overlay
I*    A     2        DC@CMN  Menu command key
I*    S     2   0    @U      Menu command key overlay
I*    A     2        DC@CDM  Messages command key
I*    S     2   0    @M      Messages command key overlay
I*    A     2        DC@CAD  Add command key
I*    S     2   0    @A      Add command key overlay
```

```
I*    A     2          DC@CCH  Change command key
I*    S     2    0     @C      Change command key overlay
I*    A     2          DC@CDL  Delete command key (@D overlay)
I*    S     2    0     @D      Delete command key overlay
I*
I*    A     1          DC@SUI  Unique system prefix / identifier
I*
I*    A    30          DC@SIZ  Standard file size
I*    A     4          DC@LCK  System file level check standard
I*
I*    A     1          DC@CSL  Produce source listings default (Y or N)
I*    A     1          DC@COP  Optimize program compiles       (Y or N)
I*    A     1          DC@CID  Ignore decimal data errors      (Y or N)
I*
I*    S     3    0     DC@DPW  Default printer width (80 - 198)
I*    S     3    0     DC@DSW  Default screen width  (80 - 132)
I*    S     3    0     DC@DSL  Default screen length (16 - 24)
I*
I*    A    40          DC@DIA  Default input attribute list (alpha)
I*    A    40          DC@DIN  Default input attribute list (numeric)
I*    A    40          DC@DOA  Default output attribute list (alpha)
I*    A    40          DC@DON  Default output attribute list (numeric)
I*    A    21          DC@$JD  Last job description used
I*    A    21          DC@$JQ  Last job queue used
I*    A    21          DC@$OQ  Last output queue used
I*    A    10          DC@DFN  Default library name for new file def'n.
I*    A     1          DC@DCC  Default commitment control for new file
I*    A     1          DC@DAC  Default AUTOCOMMIT control for new file
I*    A     1          DC@DFE  Default FCC program editor
I*
I*    A     1          DC@OSV  Operating system version no.
I*
I*    A     2          DC@EOA  Export LANSA object authority default
I*    A     1          DC@ASP  Partitions configured in differing ASPs.
I*    A     1          DC@TCO  Turbo compiler option is enabled ("T"=Yes).
I*    A     1          DC@LDC  LANSA/DOCUMENT is enabled
I*
```

```
I*    A     1       DC@EDB  Enable debug default (Y/N)
I*    A     1       DC@EAE  Enable alternate editors for help and RDML
I*    A    10       DC@DIP  Dictionary interface program
I*    A     3       DC@DPL  Default printer form length
I*    A     3       DC@DPO  Default printer overflow line number
I*
I*    A     1       DC@IHI  Default IDENT_ATR parm to high intensity
I*    A     1       DC@IRI  Default IDENT_ATR parm to reverse image
I*    A     1       DC@IUL  Default IDENT_ATR parm to underline
I*    A     1       DC@STH  Default STD_HEAD parameter value
I*
I*    A     1       DC@URL  Use running process locks
I*    A     2       DC@LPC  Last PC installed on this system
I*    A    10       DC@HOL  Name of HANDSON pgm library (if installed)
I*    A     1       DC@LLG  Display LAMDA logo screen on entry
I*    A     1       DC@EFH  Enable field level HELP support
I*    A     2       DC@C3X  System/38 or System/3X computer
I*    A     2       DC@CEE  Execution environment if System/3X
I*    A    20       DC@W    List of 132 wide screens (20 x char 1)
I*    A     1       DC@CPR  Default for compiling a process
I*    A     1       DC@EAK  Enable attention key in system
I*    A     1       DC@LFC  Field label fill character
I*    A     1       DC@CUC  Field column heading underline character
I*    A    10       DC@C    List of valid color screen codes (10xchar)
I*    A     1       DC@FLS  Function level security is enabled
I*    A     1       DC@AGH  Automatically generate field level help
I*    A     1       DC@PKE  Prompt key is enabled by default
I*    A     2       DC@CPM  Prompt key function key (non SAA only)
I*    A     1       DC@RCL  Execute RCLRSC on exit from LANSA
I*    A     1       DC@RPL  Execute RPLLIBL on exit from LANSA
I*    A     1       DC@IOM  Release 2.5 I/O modules required
I*    A     1       DC@TRP  Target release for AS/400 compiles
I*                          and saves is *PRV (previous)
I*    A     1       DC@HCH  Horizontal print character for reports
I*    A     1       DC@VCH  Vertical print character for reports
I*    A     1       DC@ESM  External Security Matching (files)
I*    A     1       DC@EXL  3GL Pgm access to exchange list required
```

```
I*    A     2         DC@CYY  YY Compare value
I*    A     2         DC@CLE  Century if LE DC@CYY
I*    A     2         DC@CGT  Century if GT DC@CYY
I*    A     1         DC@DPC  Decimal format ('.' or ',')
I*    A     1         DC@SMG  Suppress 'Interpretive mode' message.
I*    A     1         DC@SPS  Disable process/function security checking
I*    A     1         DC@SFS  Disable file security checking
I*    A     1         DC@UPK  Execute user defined prompt key message pgm
I*    A     1         DC@EXA  EXCHANGE all fields on a prompt request
I*    A     1         DC@BAT  Battery Licence MAY Apply Flag
I*
I*    S     3   1     DC@CPF  Release level of CPF
I*    S     6   0     DC@DAT  Date in system format
I*    A     3         DC@SDF  System date format
I*    S     6   0     DC@DMY  Date in DDMMYY format
I*    S     6   0     DC@YMD  Date in YYMMDD format
I*    S     6   0     DC@MDY  Date in MMDDYY format
I*    S     5   0     DC@JUL  Date in Julian format
I*
I*    A     1         DC@JTP  Current job type (B or I)
I*    A    10         DC@JOB  Current job name
I*    A     6         DC@JNB  Current job number
I*    A    10         DC@USR  Current job user
I*    A    10         DC@OQN  Current output queue name
I*    A    10         DC@OQL  Current output queue library
I*    A    10         DC@MQN  Current message queue name
I*    A    10         DC@MQL  Current message queue library
I*    A   253         DC@LBL  Current library list (first 23)
I*    A    22         DC@LB2  Current library list (last 2)
I*    A    10         DC@GRU  Current job group user profile
I*    A    10         DC@GRO  Group owner (*USRPRF *GRPPRF)
I*    A    10         DC@GRA  Group authority (*ALL *NORMAL *NONE)
I*    A     1         DC@LDM  LANSA "system" debug mode is on
I*    A     1         DC@SYT  Reserved for internal use
I*    A     3         DC@P#I  Current partition identifier
I*    A    40         DC@P#D  Current partition description
I*    A     9         DC@P#S  Displayable product/partition
```

```
I*    A     6        DC@DEX  Displayable exit key description
I*    P   1,0        DC@LEX  Length of exit key description
I*    A     1        DC@TEX  Exit key is EXIT or SYSTEM function
I*    S   9,0        DC@OQI  Next file OPNID parameter to be used
I*
I*    A     1        SA@SAA  Current partition uses SAA/CUA standards
I*    A    10        DC@WMB  Compiler Working Member Name
I*    A     7        DC@SMB  Compiler Working Source Name
I*
I*    A     1        ML@IGM  Current machine is IGC capable (1/0)
I*    A     1        ML@IGD  Current device is IGC capable (1/0)
I*    A     8        DC@TKI  Current Task Identifier
I*
I*    A     1        DC@PCT  PC Type (W=Windows, P=PM)
I*
I*    A     1        DC@ILS  Ignore LANSA security checks (Y/N)
I*                           (External export/import)
I*                   DC@SDS  System Date Separator
I*                   DC@STS  System Time Separator
I*    A     1        DC@RTR  Function Routing in Effect Flag (Y/N)
I*
```

```
IDC@IDS      DS                          1024
I                                         1  10 DC@PGL
I                                        11  20 DC@DTL
I                                        21  30 DC@MDL
I                                        31  40 DC@OWN
I                                        41  45 DC@PRO
I                                        46  481DC@LVL
I                                        49  78 DC@ORG
I                                        79  88 DC@SEC
I                                        89  98 DC@LIC
I*
I                                        99 100 DC@CEX
I                                        99 1000@E
I                                       101 102 DC@CMN
I                                       101 1020@U
```

```
I                                      103 104 DC@CDM
I                                      103 1040@M
I                                      105 106 DC@CAD
I                                      105 1060@A
I                                      107 108 DC@CCH
I                                      107 1080@C
I                                      109 110 DC@CDL
I                                      109 1100@D
I*
I                                      111 111 DC@SUI
I*
I                                      112 141 DC@SIZ
I                                      142 145 DC@LCK
I*
I                                      146 146 DC@CSL
I                                      147 147 DC@COP
I                                      148 148 DC@CID
I*
I                                      149 1510DC@DPW
I                                      152 1540DC@DSW
I                                      155 1570DC@DSL
I*
I                                      158 197 DC@DIA
I                                      198 237 DC@DIN
I                                      238 277 DC@DOA
I                                      278 317 DC@DON
I*
I                                      318 338 DC@$JD
I                                      339 359 DC@$JQ
I                                      360 380 DC@$OQ
I                                      381 390 DC@DFN
I                                      391 391 DC@DCC
I                                      392 392 DC@DAC
I                                      393 393 DC@DFE
I*
I                                      394 394 DC@OSV
I*
```

```
I                                      395 396 DC@EOA
I                                      397 397 DC@ASP
I                                      398 398 DC@TCO
I                                      399 399 DC@LDC
I*
I                                      400 400 DC@EDB
I                                      401 401 DC@EAE
I                                      402 411 DC@DIP
I                                      412 414 DC@DPL
I                                      415 417 DC@DPO
I*
I                                      418 418 DC@IHI
I                                      419 419 DC@IRI
I                                      420 420 DC@IUL
I                                      421 421 DC@STH
I                                      422 422 DC@URL
I                                      423 424 DC@LPC
I                                      425 434 DC@HOL
I                                      435 435 DC@LLG
I                                      436 436 DC@EFH
I                                      437 438 DC@C3X
I                                      439 440 DC@CEE
I                                      441 460 DC@W
I                                      461 461 DC@CPR
I                                      462 462 DC@EAK
I                                      463 463 DC@LFC
I                                      464 464 DC@CUC
I                                      465 474 DC@C
I                                      475 475 DC@FLS
I                                      476 476 DC@AGH
I                                      477 477 DC@PKE
I                                      478 479 DC@CPM
I                                      480 480 DC@RCL
I                                      481 481 DC@RPL
I                                      482 482 DC@IOM
I                                      483 483 DC@IRP
I                                      484 484 DC@HCH
```

```
I                                        485 485 DC@VCH
I                                        486 486 DC@ESM
I                                        487 487 DC@EXL
I                                        488 489 DC@CYY
I                                        490 491 DC@CLE
I                                        492 493 DC@CGT
I                                        494 494 DC@DPC
I                                        495 495 DC@SMG
I                                        496 496 DC@SPS
I                                        497 497 DC@SFS
I                                        498 498 DC@UPK
I                                        499 499 DC@EXA
I*
I                                        500 500 DC@BAT
I*
I****** END OF DC@A01 MAPPING AREA (BYTE 500) ******
I*
I                                        501 5031DC@CPF
I                                        504 5090DC@DAT
I                                        510 512 DC@SDF
I                                        513 5180DC@DMY
I                                        513 5140DC@DAY
I                                        515 5160DC@MTH
I                                        517 5180DC@YER
I                                        519 5240DC@YMD
I                                        525 5300DC@MDY
I                                        531 5350DC@JUL
I*
I                                        536 536 DC@JTP
I                                        537 546 DC@JOB
I                                        547 552 DC@JNB
I                                        553 562 DC@USR
I                                        563 572 DC@OQN
I                                        573 582 DC@OQL
I                                        583 592 DC@MQN
I                                        593 602 DC@MQL
I                                        603 855 DC@LBL
```

```
I                                        856 877 DC@LB2
I                                        603 877 DC@B
I                                        878 887 DC@GRU
I                                        888 897 DC@GRO
I                                        898 907 DC@GRA
I*
I                                        908 908 DC@LDM
I                                        909 909 DC@SYT
I                                        910 912 DC@P#I
I                                        913 952 DC@P#D
I                                        953 961 DC@P#S
I                                        962 967 DC@DEX
I                                      P 968 9680DC@LEX
I                                        969 969 DC@TEX
I                                        970 9780DC@QQI
I*
I                                        980 980 SA@SAA
I                                        981 990 DC@WMB
I                                        991 997 DC@SMB
I*
I                                        998 998 ML@IGM
I                                        999 999 ML@IGD
I*
I                                      10011008 DC@TKI
I*
I                                      10091009 DC@PCT
I*
I                                      10101010 DC@ILS
I                                      10111011 DC@SDS
I                                      10121012 DC@STS
I                                      10141014 DC@RTR
I/EJECT
I*=====================================================
I* LANSA EXCHANGE  DATA STRUCTURE PASSED TO EVERY PROGRAM
I*=====================================================
I*
I*     DC@EDS - EXTENDED SYSTEM INFORMATION
```

```
I*     ------------------------------------
I*
I*  Type  Len  Dec   Field  Description
I*  ~~~~  ~~~  ~~~   ~~~~~  ~~~~~~~~~~~
I*    A     6        DC@LDI  Identifies LDA as being set up by LANSA
I*    A     5        DC@NXT  Next task to be performed
I*    A    10        DC@PGM  Next program to be called
I*    A     1        DC@PLT  Parameter list type for call
I*    A     1        DC@PF1  Program function flag/byte 1
I*    A     1        DC@PF2  Program function flag/byte 2
I*    A    10        DC@HLM  Higher level message collector
I*
I*    A     1        DC@OCH  Object change allowed flag
I*    A     3        DC@ODM  Object definition mode
I*    A     1        DC@OAM  Object amendment flag (full amendment)
I*
I*    A     1        DC@RET  Program suite return code
I*    A     7        DC@MID  Program suite message
I*    A   132        DC@MVR  Program suite message substitution variable
I*    A     4        ML@RLN  Repeated language for batch jobs
I*
I*    A    10        DC@CPI  Current process access key
I*    A     7        DC@CFI  Current function access key
I*    A     1        DC@OAP  Object amendment flag (partial amendment)
I*    S   3,0        DC@NUP  Number of user parameters passed
I*    A     1        DC@FUM  Current function action (C/R)
I*    A     1        DC@RTP  Request to terminate program
I*    P  15,0        DC@WRN  WITH_RRN value
I*    P  15,0        DC@RRN  RETURN_RRN value
I*    A     1        DC@DBG  DEBUG mode on for current function
I*    A    10        DC@DBP  DEBUG program name
I*    A     1        DC@DCI  Direct call indicator
I*    A     1        DC@CWF  Compiler work files in use flag
I*    A     1        DC@ACC  Auto commitment control requested flag
I*    A     3        DC@E#I  Submitting partition identifier
I*    A     1        DC@WIU  Wide screen currently in use flag
I*
```

```
I* Permanent fields beyond the overlay area ......
I*
I*    A    10        DC@CPN  Current process name
I*    A     7        DC@LFN  Last function name
I*    A     7        DC@CFN  Current function name
I*    A     7        DC@NFN  Next function name
I*    A    10        DC@DCF  Direct call function name
I*    A     1        DC@WIS  Wide screen indicator ('Y' 'N')
I*    A    10        DC@D$P  Process to be de-bugged
I*    A     7        DC@D$F  Function to be de-bugged
I*    A     1        DC@D$E  Debug currently enabled for a function
I*    P   3,0        DC@CFS  Current function DC@F29/30 set number
I*    A     1        DC@JCC  Optional job completion code (Y or N)
I*    A     5        DC@SXT  Suspended next task to be performed
I*    A    10        DC@SGM  Suspended next program to be called
I*    A    10        DC@L$P  Locked process name
I*    A    10        DC@L$F  Locked function name
I*    A     1        DC@MSG  Work with Developer Messages (Y/N)
I*    A     1        DC@DEV  Development User (Y/N)
I*
I* SAA/CUA DETAILS
I*
I*    A     1        SA@PMS  Process menu style (N=Number,C=Cursor)
I*    A    76        SA@PMP  Process menu prompt line
I*    A     1        SA@PTU  Process title in uppercase (Y,N)
I*    A     1        SA@PTU  Function title in uppercase (Y,N)
I*    A     1        SA@ABS  Display panel identifier
I*    S   2,0        SA@ABP  ** Spare **
I*    S   2,0        SA@ABW  ** Spare **
I*    A     1        SA@PDS  ** Spare **
I*    S   2,0        SA@MSL  Message line number (22,23,24)
I*    S   2,0        SA@F1L  Function key line number - 1 (22,23,24)
I*    S   2,0        SA@F2L  Function key line number - 2 (22,23,24)
I*                           Common dialog actions function keys
I*    A     2        SA@KCN  Cancel (LANSA menu key)
I*    A     2        SA@KEN  Enter
I*    A     2        SA@KHP  Help
```

```
I*   A     2        SA@KPR  Prompt (optional)
I*   A     2        SA@KRF  Refresh (optional)
I*   A     2        SA@KMS  Messages (LANSA key)
I*                          Other dialog actions function keys
I*   A     2        SA@KBK  Backward
I*   A     2        SA@KCM  Command (optional)
I*   A     2        SA@KDK  Display keys (optional)
I*   A     2        SA@KEH  Exit high (application level)
I*                          (LANSA exit key)
I*   A     2        SA@KEL  Exit low (function level)
I*   A     2        SA@KHM  Home
I*   A     2        SA@KFW  Forward (optional)
I*   A     2        SA@KHC  Help contents
I*   A     2        SA@KHE  Extended help
I*   A     2        SA@KHI  Help index
I*   A     2        SA@KHK  Keys help
I*   A     2        SA@KLF  Left (optional)
I*   A     2        SA@KRE  Retrieve (optional)
I*   A     2        SA@KRT  Right (optional)
I*   A     2        SA@KSB  Switch backward (optional)
I*   A     2        SA@KSF  Switch forward (optional)
I*   A     2        SA@KSA  Switch to action bar (optional)
I*   A     2        SA@KAD  Add (LANSA key)
I*   A     2        SA@KCH  Change (LANSA key)
I*   A     2        SA@KDL  Delete (LANSA key)
I*                   SA@K    Array to overlay function keys
I*                           Common function keys description
I*   A    10        SA@DCN  Cancel
I*   A    10        SA@DEN  Enter
I*   A    10        SA@DHP  Help
I*   A    10        SA@DPR  Prompt (optional)
I*   A    10        SA@SRF  Refresh (optional)
I*                           Other function keys description
I*   A    10        SA@SBK  Backward
I*   A    10        SA@DCM  Command (optional)
I*   A    10        SA@DDK  Display keys (optional)
I*   A    10        SA@DEH  Exit high (application level)
```

```
I*    A   10        SA@DEL  Exit low (function level)
I*    A   10        SA@DHM  Home
I*    A   10        SA@DFW  Forward (optional)
I*    A   10        SA@DHC  Help contents
I*    A   10        SA@DHE  Extended help
I*    A   10        SA@DHI  Help index
I*    A   10        SA@DHK  Keys help
I*    A   10        SA@DLF  Left (optional)
I*    A   10        SA@DRE  Retrieve (optional)
I*    A   10        SA@DRT  Right (optional)
I*    A   10        SA@DSB  Switch backward (optional)
I*    A   10        SA@DSF  Switch forward (optional)
I*    A   10        SA@DSA  Switch to action bar (optional)
I*    A   10        SA@DAD  Add (LANSA key)
I*    A   10        SA@DCH  Change (LANSA key)
I*    A   10        SA@DDL  Delete (LANSA key)
I*                  SA@D    Array to overlay fn. key description
I*                          Common function keys desc. length
I*    S   2,0       SA@LCN  Cancel
I*    S   2,0       SA@LEN  Enter
I*    S   2,0       SA@LHP  Help
I*    S   2,0       SA@LPR  Prompt (optional)
I*    S   2,0       SA@LRF  Refresh (optional)
I*                          Other function keys desc. length
I*    S   2,0       SA@LBK  Backward
I*    S   2,0       SA@LCM  Command (optional)
I*    S   2,0       SA@LDK  Display keys (optional)
I*    S   2,0       SA@LEH  Exit high (application level)
I*    S   2,0       SA@LEL  Exit low (function level)
I*    S   2,0       SA@LHM  Home
I*    S   2 0       SA@LFW  Forward (optional)
I*    S   2,0       SA@LHC  Help contents
I*    S   2,0       SA@LHE  Extended help
I*    S   2,0       SA@LHI  Help index
I*    S   2,0       SA@LHK  Keys help
I*    S   2,0       SA@LLF  Left (optional)
I*    S   2,0       SA@LRE  Retrieve (optional)
```

```
I*    S   2,0        SA@LRT  Right (optional)
I*    S   2,0        SA@LSB  Switch backward (optional)
I*    S   2,0        SA@LSF  Switch forward (optional)
I*    S   2,0        SA@LSA  Switch to action bar (optional)
I*    S   2,0        SA@LAD  Add (LANSA key)
I*    S   2,0        SA@LCH  Change (LANSA key)
I*    S   2,0        SA@LDL  Delete (LANSA key)
I*                   SA@L    Array to overlay fn. key desc. length
I*    A     1        SA@PDT  Show process date/time
I*    A     1        SA@FDT  Show function date/time
I*    A     1        SA@PAR  Process auto-record advance
I*    A     1        SA@FAR  Function auto-record advance
I*    A     1        SA@AAR  Action bar auto-record advance
I*
I*    A     5        DC@LIO  Last file I/O status code
I*
I* New multilingual flags
I*
I*    A     1        ML@CNV  Multilingual - IGCCNV required (Y/N)
I*    A     2        ML@KCV  Multilingual - IGCCNV function key
I*    A    10        ML@DCV  Multilingual - IGCCNV key description
I*    S   2,0        ML@LDC  Multilingual - IGCCNV key desc. length
I*    A     2        ML@LCV  Multilingual - IGCCNV line number
I*    A     1        ML@DEV  Multilingual development language (Y/N)
I*    A   100        ML@MSG  Multilingual message files
I*    A     1        DC@MLP  Multilingual partition indicator (Y/N)
I*    A     4        ML@CLN  Current language code
I*    A    20        ML@CLD  Current language description
I*    A     1        ML@LRL  Current language is left-to-right
I*    A     1        ML@IGL  Current language is IGC / DBCS
I*    A     1        ML@RLL  Current language is right-to-left
I*    A     4        ML@DLN  Default development language
I*    P   3,0        ML@LRT  Total left-to-right languages in partition
I*    P   3,0        ML@IGT  Total IGC / DBCS languages in partition
I*    P   3,0        ML@RLT  Total right-to-left languages in partition
I*
I* New partition level details
```

```
I*
I*    A     40        P#@PHM  Process menu HELP message
I*    A     40        P#@PRM  Process menu "RETURN TO" message prefix
I*    A     40        P#@PEM  Process menu EXIT message
I*
I*    A     1         P#@ILL  Insert default file library in library list
I*
I*    A     1         P#@KRG  Keep RPG source statements
I*    A     10        P#@FRG  Keep RPG source file
I*    A     10        P#@LRG  Keep RPG source library
I*
I*    A     1         P#@KDD  Keep DDS source statements
I*    A     10        P#@FDD  Keep DDS source file
I*    A     10        P#@LDD  Keep DDS source library
I*
I*                            Common fn. keys enabled ('1' '0')
I*    A     1         SA@ECN  Cancel (LANSA menu key)
I*    A     1         SA@EEN  Enter
I*    A     1         SA@EHP  Help
I*    A     1         SA@EPR  Prompt (optional)
I*    A     1         SA@ERF  Refresh (optional)
I*    A     1         SA@EMS  Messages (LANSA key)
I*                            Other dialog actions function keys
I*    A     1         SA@EBK  Backward
I*    A     1         SA@ECM  Command (optional)
I*    A     1         SA@EDK  Display keys (optional)
I*    A     1         SA@EEH  Exit high (application level)
I*    A     1                 (LANSA exit key)
I*    A     1         SA@EEL  Exit low (function level)
I*    A     1         SA@EHM  Home
I*    A     1         SA@EFW  Forward (optional)
I*    A     1         SA@EHC  Help contents
I*    A     1         SA@EHE  Extended help
I*    A     1         SA@EHI  Help index
I*    A     1         SA@EHK  Keys help
I*    A     1         SA@ELF  Left (optional)
I*    A     1         SA@ERE  Retrieve (optional)
```

```
I*    A     1        SA@ERT  Right (optional)
I*    A     1        SA@ESB  Switch backward (optional)
I*    A     1        SA@ESF  Switch forward (optional)
I*    A     1        SA@ESA  Switch to action bar (optional)
I*    A     1        SA@EAD  Add (LANSA key)
I*    A     1        SA@ECH  Change (LANSA key)
I*    A     1        SA@EDL  Delete (LANSA key)
I*                   SA@E    Array to overlay SA@Exx
I*                   SA@EXX  Field to overlay SA@Exx
I*
I*    A     1        SA@DTC  Display type code (ex file IDS)
I*    A     1        SA@COL  Color display device ('1' '0')
I*
IDC@EDS      DS                          2500
I                                         1   6 DC@IDN
I                                         7  11 DC@NXT
I                                        12  21 DC@PGM
I                                        22  22 DC@PLT
I                                        23  23 DC@PF1
I                                        24  24 DC@PF2
I                                        25  34 DC@HLM
I*
I                                        35  35 DC@OCH
I                                        36  36 DC@ODM
I                                        37  37 DC@OAM
I*
I                                        38  38 DC@RET
I                                        39  45 DC@MID
I                                        46 177 DC@MVR
I                                        46 177 MVR
I*
I                                       174 177 ML@RLN
I*
I                                       178 187 DC@CPI
I                                       188 194 DC@CFI
I                                       195 195 DC@OAP
I                                       196 1980DC@NUP
```

```
I                                      199 199 DC@FUM
I                                      200 200 DC@RTP
I                                    P 201 2080DC@WRN
I                                    P 209 2160DC@RRN
I                                      217 217 DC@DBG
I                                      218 227 DC@DBP
I                                      228 228 DC@DCI
I                                      229 229 DC@CWF
I                                      232 232 DC@ACC
I                                      233 235 DC@E#I
I                                      236 236 DC@WIU
I*
I* SAA/CUA details and other partition level information from F46IDS
I*
I                                   10011250 SA@I01
I                                   12511500 SA@I02
I                                   15011750 SA@I03
I                                   17512000 SA@I04
I*
I                                   10011001 SA@PMS
I                                   10021077 SA@PMP
I                                   10781078 SA@PTU
I                                   10791079 SA@FTU
I                                   10801080 SA@ABS
I                                   108110820SA@ABP
I                                   108310840SA@ABW
I                                   10851085 SA@PDS
I                                   108610870SA@MSL
I                                   108810890SA@F1L
I                                   109010910SA@F2L
I*
I                                   10921093 SA@KCN
I                                   10941095 SA@KEN
I                                   10961097 SA@KHP
I                                   10981099 SA@KPR
I                                   11001101 SA@KRF
I                                   11021103 SA@KMS
```

```
I                                    11041105 SA@KBK
I                                    11061107 SA@KCM
I                                    11081109 SA@KDK
I                                    11101111 SA@KEH
I                                    11121113 SA@KEL
I                                    11141115 SA@KHM
I                                    11161117 SA@KFW
I                                    11181119 SA@KHC
I                                    11201121 SA@KHE
I                                    11221123 SA@KHI
I                                    11241125 SA@KHK
I                                    11261127 SA@KLF
I                                    11281129 SA@KRE
I                                    11301131 SA@KRT
I                                    11321133 SA@KSB
I                                    11341135 SA@KSF
I                                    11361137 SA@KSA
I                                    11381139 SA@KAD
I                                    11401141 SA@KCH
I                                    11421143 SA@KDL
I                                    10921151 SA@K
I*
I                                    11521161 SA@DCN
I                                    11621171 SA@DEN
I                                    11721181 SA@DHP
I                                    11821191 SA@DPR
I                                    11921201 SA@DRF
I                                    12021211 SA@DMS
I                                    12121221 SA@DBK
I                                    12221231 SA@DCM
I                                    12321241 SA@DDK
I                                    12421251 SA@DEH
I                                    12521261 SA@DEL
I                                    12621271 SA@DHM
I                                    12721281 SA@DFW
I                                    12821291 SA@DHC
I                                    12921301 SA@DHE
```

```
I                                      13021311 SA@DHI
I                                      13121321 SA@DHK
I                                      13221331 SA@DLF
I                                      13321341 SA@DRE
I                                      13421351 SA@DRT
I                                      13521361 SA@DSB
I                                      13621371 SA@DSF
I                                      13721381 SA@DSA
I                                      13821391 SA@DAD
I                                      13921401 SA@DCH
I                                      14021411 SA@DDL
I                                      11521451 SA@D
I*
I                                     145214530SA@LCN
I                                     145414550SA@LEN
I                                     145614570SA@LHP
I                                     145814590SA@LPR
I                                     146014610SA@LRF
I                                     146214630SA@LMS
I                                     146414650SA@LBK
I                                     146614670SA@LCM
I                                     146814690SA@LDK
I                                     147014710SA@LEH
I                                     147214730SA@LEL
I                                     147414750SA@LHM
I                                     147614770SA@LFW
I                                     147814790SA@LHC
I                                     148014810SA@LHE
I                                     148214830SA@LHI
I                                     148414850SA@LHK
I                                     148614870SA@LLF
I                                     148814890SA@LRE
I                                     149014910SA@LRT
I                                     149214930SA@LSB
I                                     149414950SA@LSF
I                                     149614970SA@LSA
I                                     149814990SA@LAD
```

```
I                                      150015010SA@LCH
I                                      150215030SA@LDL
I                                       14521511 SA@L
I                                       14521511 CHSA@L
I*
I                                       15121512 SA@PDT
I                                       15131513 SA@FDT
I                                       15141514 SA@PAR
I                                       15151515 SA@FAR
I                                       15161516 SA@AAR
I*
I                                       15731577 DC@LIO
I*
I* New multilingual flags (EX F46IDS & F60IDS)
I*
I                                      168016810ML@LDC
I                                       16821682 ML@CNV
I                                       16831684 ML@KCV
I                                       16851694 ML@DCV
I                                       16951696 ML@LCV
I                                       16971697 ML@DEV
I                                       16981797 ML@MSG
I                                       17981798 DC@MLP
I                                       17991802 ML@CLN
I                                       18031822 ML@CLD
I                                       18231823 ML@LRL
I                                       18241824 ML@IGL
I                                       18251825 ML@RLL
I                                       18261829 ML@DLN
I                                      P183018310ML@LRT
I                                      P183218330ML@IGT
I                                      P183418350ML@RLT
I*
I* New partition level details (EX F46IDS)
I*
I                                       18381877 P#@PHM
I                                       18781917 P#@PRM
```

```
I                                      19181957 P#@PEM
I                                      19581958 P#@ILL
I                                      19591959 P#@KRG
I                                      19601969 P#@FRG
I                                      19701979 P#@LRG
I                                      19801980 P#@KDD
I                                      19811990 P#@FDD
I                                      19912000 P#@LDD
I*
I                                      21592159 SA@ECN
I                                      21602160 SA@EEN
I                                      21612161 SA@EHP
I                                      21622162 SA@EPR
I                                      21632163 SA@ERF
I                                      21642164 SA@EMS
I                                      21652165 SA@EBK
I                                      21662166 SA@ECM
I                                      21672167 SA@EDK
I                                      21682168 SA@EEH
I                                      21692169 SA@EEL
I                                      21702170 SA@EHM
I                                      21712171 SA@EFW
I                                      21722172 SA@EHC
I                                      21732173 SA@EHE
I                                      21742174 SA@EHI
I                                      21752175 SA@EHK
I                                      21762176 SA@ELF
I                                      21772177 SA@ERE
I                                      21782178 SA@ERT
I                                      21792179 SA@ESB
I                                      21802180 SA@ESF
I                                      21812181 SA@ESA
I                                      21822182 SA@EAD
I                                      21832183 SA@ECH
I                                      21842184 SA@EDL
I                                      21592188 SA@E
I                                      21592188 SA@EXX
```

```
I*
I                                        22202220 SA@DTC
I                                        22212221 SA@COL
```

PR@IDS - Process Information

The /COPY source member PR@ISPEC can be found in file DC@F28 in the LANSA Data Library.

```
I*=======================================================
I* LANSA INFORMATION DATA STRUCTURE PASSED TO EVERY PROGRAM
I*=======================================================
I*
I* PR@IDS - PROCESS INFORMATION DATA STRUCTURE
I*
I*  Type  Len  Dec   Field  Description
I*  ~~~~  ~~~  ~~~   ~~~~~  ~~~~~~~~~~~
I*    A     7        PR@FUN  NAME OF CURRENT FUNCTION
I*    A    40        PR@FUD  CURRENT FUNCTION DESCRIPTION
I*    A     7        PR@NXT  NAME OF NEXT FUNCTION
I*    A    40        PR@NXD  NEXT FUNCTION DESCRIPTION
I*    A     3        PR@TYP  PROCESS TYPE
I*    A    10        PR@PRO  NAME OF CURRENT PROCESS
I*    A    40        PR@PRD  CURRENT PROCESS DESCRIPTION
I*    A     1        PR@RCL  FUNCTION SHOULD RECLAIM RESOURCES (Y OR N)
I*    A     3        PR@MDE  CURRENT PROCESS MODE
I*    A     1        PR@VER  VALIDATION ERROR FLAG
I*
I*    A     2        PR@CEX  Exit command key
I*    A     2        PR@CMN  Menu command key
I*    A     2        PR@CDM  Messages command key
I*    A     2        PR@CAD  Add command key
I*    A     2        PR@CCH  Change command key
I*    A     2        PR@CDL  Delete command key (@D overlay)
I*
I*    Array 5*2      PR@C    All user command keys
I*    A    10        PR@CUK  All user command keys
```

```
I*    A      2        PR@CU1  User command key 1
I*    A      2        PR@CU2  User command key 2
I*    A      2        PR@CU3  User command key 3
I*    A      2        PR@CU4  User command key 4
I*    A      2        PR@CU5  User command key 5
I*
I*    Array 5*8       PR@D    All command key decsriptions
I*    A      8        PR@CD1  Description of user key 1
I*    A      8        PR@CD2  Description of user key 2
I*    A      8        PR@CD3  Description of user key 3
I*    A      8        PR@CD4  Description of user key 4
I*    A      8        PR@CD5  Description of user key 5
I*
I*    A      2        PR@KEY  Last command key hit by user
I*
I*    A      6        PR@ELK  Enable/disable all keys control
I*    A      1        PR@EEX  Enable EXIT key control
I*    A      1        PR@EMN  Enable MENU key control
I*    A      1        PR@EDM  Enable MESSAGES key control
I*    A      1        PR@EAD  Enable ADD key control
I*    A      1        PR@ECH  Enable CHANGE key control
I*    A      1        PR@EDL  Enable DELETE key control
I*
I*    S    3,0        PR@B@N  Built-In Function unique identifier
I*    A      1        PR@B@A  Built-In Function action required
I*
I*    A      1        PR@EPR  Enable PROMPT key control
I*
I*    A      1        PR@MUS  PROCESS MENU STYLE
I*    A      3        PR@HLT  HELP TYPE REQUIRED
I*    A     10        PR@PHP  CURRENT PROCESS HELP POINTER
I*    A     10        PR@FHP  CURRENT FUNCTION HELP POINTER
I*    A     10        PR@RMQ  RETURN MESSAGE QUEUE (5001)
I*
I*    A      1        PR@UAC  UNFORMATTED ROUTINE ACTION CODE
I*    A      2        PR@URC  UNFORMATTED ROUTINE RETURN CODE
I*    A     10        PR@UMQ  UNFORMATTED ROUTINE MESSAGE QUEUE
```

```
I*    A     1        PR@UEA  UNFORMATTED ROUTINE EXIT ENABLED
I*    A     1        PR@UMA  UNFORMATTED ROUTINE MENU ENABLED
I*    A     1        PR@UAA  UNFORMATTED ROUTINE ADD ENABLED
I*    A     1        PR@UCA  UNFORMATTED ROUTINE CHANGE ENABLED
I*    A     1        PR@UDA  UNFORMATTED ROUTINE DELETE ENABLED
I*
I*    P   3,0        PR@RWI  REPORT WIDTH TO PRINT
I*    P   1,0        PR@RSP  REPORT NUMBER OF SPACES
I*    A     1        PR@RNO  REPORT NUMBER
I*    A    40        PR@RTL  REPORT TITLE
I*    P   3,0        PR@RSE  REPORT COLUMN SEPERATION
I*    P   3,0        PR@RSC  REPORT START COLUMN NUMBER
I*    A     7        PR@MDZ  CURRENT PROCESS MODE DESCRIPTION
I*    A    10        PR@PGM  CURRENT PROCESS PROGRAM NAME
I*
I*    A  50*7        PR@A    ALLOWABLE NEXT FUNCTIONS LIST
I*    A   175        PR@A01  ALLOWABLE NEXT FUNCTIONS LIST - PART 1
I*    A   175        PR@A02  ALLOWABLE NEXT FUNCTIONS LIST - PART 2
I*
I*    A    78        PR@CKL  ENABLED COMMAND KEY DISPLAY LINE
I*    A    24        PR@CKE  COMMAND KEY ENABLEMENT CONTROL
I*    A    10        PR@NIC  NEXT 5001 SERIES CONTROLLER PROGRAM
I*    A     1        PR@RQR  5001 RECURSIVE SCREEN RE-BUILD REQUIRED
I*    A     2        PR@RKE  ROLL KEY ENABLEMENT CONTROL
I*
I*    A     1        PR@SAA  PROCESS USES SAA/CUA STANDARDS
I*
I*    A     1        PR@OVR  ALWAYS "N" (REDUNDANT FROM RELEASE 2.5)
I*
I*    P   5,0        PR@NXL  NUMBER OF FIELDS IN PR@X
I*    P   5,0        PR@NXP  NEXT FREE POSITION IN PR@X
I*    A 2000*1       PR@X    FIELD EXCHANGE LIST
I*
I*    A  256*1       PR@G    DEBUG POINTS
I*    A   256        PR@DBG  DEBUG POINTS
I*
IPR@IDS      DS                          5000
```

```
I                                     1   7 PR@FUN
I                                    11  50 PR@FUD
I                                    51  57 PR@NXT
I                                    61 100 PR@NXD
I*
I                                   101 110 PR@C
I                                   101 110 PR@CUK
I                                   101 102 PR@CU1
I                                   103 104 PR@CU2
I                                   105 106 PR@CU3
I                                   107 108 PR@CU4
I                                   109 110 PR@CU5
I                                   111 150 PR@D
I                                   111 118 PR@CD1
I                                   119 126 PR@CD2
I                                   127 134 PR@CD3
I                                   135 142 PR@CD4
I                                   143 150 PR@CD5
I*
I                                   151 152 PR@KEY
I*
I                                   151 156 PR@ELK
I                                   151 151 PR@EEX
I                                   152 152 PR@EMN
I                                   153 153 PR@EDM
I                                   154 154 PR@EAD
I                                   155 155 PR@ECH
I                                   156 156 PR@EDL
I*
I                                   157 1590PR@B@N
I                                   160 160 PR@B@A
I*
I                                   161 161 PR@EPR
I*
I                                   241 243 PR@TYP
I                                   244 253 PR@PRO
I                                   254 293 PR@PRD
```

```
I                                    294 294 PR@RCL
I                                    295 297 PR@MDE
I                                    298 298 PR@VER
I*
I                                    299 300 PR@CEX
I                                    301 302 PR@CMN
I                                    303 304 PR@CDM
I                                    305 306 PR@CAD
I                                    307 308 PR@CCH
I                                    309 310 PR@CDL
I*
I                                    311 311 PR@MUS
I                                    318 320 PR@HLT
I                                    321 330 PR@PHP
I                                    331 340 PR@FHP
I                                    341 350 PR@RMQ
I*
I                                    351 351 PR@UAC
I                                    352 353 PR@URC
I                                    354 363 PR@UMQ
I                                    364 364 PR@UEA
I                                    365 365 PR@UMA
I                                    366 366 PR@UAA
I                                    367 367 PR@UCA
I                                    368 368 PR@UDA
I*
I                                  P 369 3700PR@RWI
I                                  P 371 3710PR@RSP
I                                    372 372 PR@RNO
I                                    373 412 PR@RTL
I                                  P 413 4140PR@RSE
I                                  P 415 4160PR@RSC
I                                    417 423 PR@MDZ
I                                    424 433 PR@PGM
I*
I                                    501 850 PR@A
I                                    501 675 PR@A01
```

```
I                                    676 850 PR@A02
I*
I                                    851 928 PR@CKL
I                                    929 952 PR@CKE
I                                    953 962 PR@NIC
I                                    961 9620PR@NI2
I                                    963 963 PR@RQR
I                                    964 965 PR@RKE
I                                    966 966 PR@SAA
I*
I                                    998 998 PR@OVR
I*
I                                  P250125030PR@NXL
I                                  P250425060PR@NXP
I                                   25074506 PR@X
I                                   25072706 PR@X01
I                                   27072906 PR@X02
I                                   29073106 PR@X03
I                                   31073306 PR@X04
I                                   33073506 PR@X05
I                                   35073706 PR@X06
I                                   37073906 PR@X07
I                                   39074106 PR@X08
I                                   41074306 PR@X09
I                                   43074506 PR@X10
I*
I                                   45074762 PR@G
I                                   45074762 PR@DBG
```

Sample Set up of Process Information - PR@IDS

The following RPG/400 code shows the necessary field settings for the Process Information data structure - PR@IDS.

```
I* Program Information Data Structure
I*
I$PIDSDS200
I*PROGRAM $PGMNM
I*
C* Clear the exchange list
C*
CZ-ADD0PR@NXL
CZ-ADD1PR@NXP
C*
CMOVEL<function>PR@FUNFunction name
CMOVEL<fdesc>PR@FUDFunction description
CMOVEL*BLANKSPR@NXT
CMOVEL*BLANKSPR@NXD
CMOVEL<ptype>PR@TYPProcess type MNU,FUN
c*CMD, HLP, or EXT
CMOVEL<process> PR@PROProcess name
CMOVEL<pdesc>PR@PRDProcess description
CMOVEL'L'PR@RCLHeavy = N, LIGHT = L
CMOVEL'DIS'PR@MDE
CMOVEL*BLANKSPR@VER
CMOVELSA@KEHPR@CEX
CMOVELSA@KCNPR@CMN
CMOVELSA@KMSPR@CDM
CMOVELSA@KADPR@CAD
CMOVELSA@KCHPR@CCH
CMOVELSA@KDLPR@CDL
CMOVEL'00'PR@CU1
CMOVEL'00'PR@CU2
CMOVEL'00'PR@CU3
CMOVEL'00'PR@CU4
CMOVEL'00'PR@CU5
```

```
CMOVE *BLANKSPR@CD1
CMOVE *BLANKSPR@CD2
CMOVE *BLANKSPR@CD3
CMOVE *BLANKSPR@CD4
CMOVE *BLANKSPR@CD5
CMOVEL*BLANKSPR@KEY
CMOVEL*BLANKSPR@ELK
C*
C*
C* $PGMNM is the name of this program
C* from the Program Information Data Structure (PIDS)
C*
CMOVEL$PGMNMPR@RMQ
CMOVE '01'PR@NIC
CMOVEL'Y'PR@RQR
CMOVEL'Y'PR@SAA
CMOVE 'N'PR@EPR
```

Programs UD@CALL1 and UD@CALL2

Program UD@CALL1

The source member for this program (UD@CALL1) can be found in file DC@F28 in the LANSA Data Library.

NOTE : Only the program source is supplied. The program object will need to be created before it can be executed.

```
/*    %NATIVE                                                        */
/*    %AS    DLTPGM     PGM($$CRTLIB$$/UD@CALL1)                    */
/*    %AS    CRTCLPGM   PGM($$CRTLIB$$/UD@CALL1) SRCFILE(DCSRC) +   */
/*    %AS               USRPRF(*OWNER) LOG(*NO) ALWRTVSRC(*NO) +    */
/*    %AS               AUT(*USE) SRCMBR(H3UD@CALL1) TGTRLS(*PRV)   */
/*    %AS    RMVM       FILE(DC@F28) MBR(UD@CALL1)                  */
/*    %AS    CPYF       FROMFILE(DC@SRCLIB/DCSRC) TOFILE(DC@F28) +  */
/*    %AS               FROMMBR(H3UD@CALL1) TOMBR(UD@CALL1) +       */
/*    %AS               MBROPT(*REPLACE)                            */
/*============================================================*/
/*                                                                */
/*  Program      : UD@CALL1                                       */
/*  ~~~~~~~                                                       */
/*  Date written : 16th APRIL 1992                                */
/*  ~~~~~~~~~~~~                                                  */
/*  Author       : Ian Smith                                      */
/*  ~~~~~~                                                        */
/*  Description  : This program is used to set up                 */
/*  ~~~~~~~~~~~    DC@IDS - System Information Data Structure     */
/*                 for use by 3GL programs to directly invoke     */
/*                 LANSA functions.                               */
/*                                                                */
/*============================================================*/
```

```
/*                  Program Amendment History                          */
/*=====================================================================*/
/*                                                                      +
   Ref    Date      Amendor Name   Brief Description of Amendment       +
   ---    ----      ------------   ------------------------------       +
   NNNNN  DD/DD/DD  XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  +
   NNNNN  DD/DD/DD  XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  +
   NNNNN  DD/DD/DD  XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  +
                                                                       */
/*=====================================================================*/
UD@CALL1:   PGM        PARM(&UD@PGL &UD@P#I &UD@LNG &DCIDS &DCEDS)
/*=====================================================================*/
/* Declare all variables ......                                        */
/*=====================================================================*/

     DCL        VAR(&UD@PGL) TYPE(*CHAR) LEN(10)
                /* LANSA Program Library */
     DCL        VAR(&UD@P#I) TYPE(*CHAR) LEN(3)
                /* LANSA Partition */
     DCL        VAR(&UD@LNG) TYPE(*CHAR) LEN(4)
                /* Language Code */
     /* The language code for the language you want to run +
        the application under. EG ENG, FRA, DUE             */
     DCL        VAR(&DCIDS) TYPE(*CHAR) LEN(1024)
                /* System Information */
     DCL        VAR(&DCEDS) TYPE(*CHAR) LEN(2500)
                /* Extended System Information */

     DCL        VAR(&DTALIB) TYPE(*CHAR) LEN(10)
     DCL        VAR(&DCPROP) TYPE(*CHAR) LEN(10)
     DCL        VAR(&DC@JTP) TYPE(*CHAR) LEN(1)
     DCL        VAR(&DC@JOB) TYPE(*CHAR) LEN(10)
     DCL        VAR(&DC@JNB) TYPE(*CHAR) LEN(6)
     DCL        VAR(&DC@USR) TYPE(*CHAR) LEN(10)
     DCL        VAR(&DC@OQN) TYPE(*CHAR) LEN(10)
     DCL        VAR(&DC@OQL) TYPE(*CHAR) LEN(10)
     DCL        VAR(&DC@MQN) TYPE(*CHAR) LEN(10)
```

```
DCL        VAR(&DC@MQL) TYPE(*CHAR) LEN(10)
DCL        VAR(&DC@LBL) TYPE(*CHAR) LEN(275)
DCL        VAR(&CPFLVL) TYPE(*CHAR) LEN(6)
DCL        VAR(&DC@DAT) TYPE(*CHAR) LEN(6)
DCL        VAR(&DC@SDF) TYPE(*CHAR) LEN(3)
DCL        VAR(&DC@DMY) TYPE(*CHAR) LEN(6)
DCL        VAR(&DC@YMD) TYPE(*CHAR) LEN(6)
DCL        VAR(&DC@MDY) TYPE(*CHAR) LEN(6)
DCL        VAR(&DC@JUL) TYPE(*CHAR) LEN(5)
DCL        VAR(&DC@LIC) TYPE(*CHAR) LEN(10)
DCL        VAR(&LDAIDN) TYPE(*CHAR) LEN(6)
DCL        VAR(&OK) TYPE(*CHAR) LEN(1)
DCL        VAR(&JOBD) TYPE(*CHAR) LEN(10)
DCL        VAR(&JOBDLIB) TYPE(*CHAR) LEN(10)
DCL        VAR(&DC@GRU) TYPE(*CHAR) LEN(10)
DCL        VAR(&DC@GRO) TYPE(*CHAR) LEN(10)
DCL        VAR(&DC@GRA) TYPE(*CHAR) LEN(10)
DCL        VAR(&OUTQ) TYPE(*CHAR) LEN(10)
DCL        VAR(&OUTQLIB) TYPE(*CHAR) LEN(10)
DCL        VAR(&DC@C3X) TYPE(*CHAR) LEN(2)
DCL        VAR(&DC@CEE) TYPE(*CHAR) LEN(2)
DCL        VAR(&DC@RCL) TYPE(*CHAR) LEN(1)
DCL        VAR(&DC@RPL) TYPE(*CHAR) LEN(1)
DCL        VAR(&ML@IGM) TYPE(*CHAR) LEN(1)
DCL        VAR(&DC@BAT) TYPE(*CHAR) LEN(1)
DCL        VAR(&MBRNAM) TYPE(*CHAR) LEN(10)
DCL        VAR(&DC@OSV) TYPE(*CHAR) LEN(1)
DCL        VAR(&RESULT) TYPE(*CHAR) LEN(22)
DCL        VAR(&RSTLEN) TYPE(*CHAR) LEN(4)
DCL        VAR(&FORMAT) TYPE(*CHAR) LEN(8) +
             VALUE('PRDR0700')
DCL        VAR(&PRODUCT) TYPE(*CHAR) LEN(27) +
             VALUE('*OPSYS *CUR  0000*CODE     ')
DCL        VAR(&ERRCDE) TYPE(*CHAR) LEN(4)

DCL        VAR(&DC@SDS) TYPE(*CHAR) LEN(1)
DCL        VAR(&DC@STS) TYPE(*CHAR) LEN(1)
```

```
/*============================================================*/
/* Global error handler ......                                */
/*============================================================*/

     MONMSG (CPF0000 MCH0000 CRG0000 DCM0000) EXEC(GOTO ERROR)

/*============================================================*/
/* Program mainline .....                                     */
/*============================================================*/

/* Retrieve the system definition data area                   */

      RTVDTAARA  DTAARA(&UD@PGL/DC@A01) RTNVAR(&DCIDS)
       CHGVAR     VAR(&UD@PGL) VALUE(%SUBSTRING(&DCIDS 1 10))
      CHGVAR     VAR(&DTALIB) VALUE(%SUBSTRING(&DCIDS 11 10))
      CHGVAR     VAR(&DC@C3X) VALUE(%SUBSTRING(&DCIDS 437 2))
      CHGVAR     VAR(&DC@CEE) VALUE(%SUBSTRING(&DCIDS 439 2))
      CHGVAR     VAR(&DC@RCL) VALUE(%SUBSTRING(&DCIDS 480 1))
      CHGVAR     VAR(&DC@RPL) VALUE(%SUBSTRING(&DCIDS 481 1))
      CHGVAR     VAR(&DC@BAT) VALUE(%SUBSTRING(&DCIDS 500 1))

      CHGVAR     VAR(&DCPROP) VALUE(%SUBSTRING(&DCIDS 41 5))
      CHGVAR     VAR(&DC@LIC) VALUE(%SUBSTRING(&DCIDS 89 10))
      CHGVAR     VAR(&LDAIDN) VALUE(X'FF')
      CHGVAR     VAR(%SUBSTRING(&LDAIDN 2 5)) +
                   VALUE(%SUBSTRING(&DCIDS 41 5))

      CHGVAR     VAR(&DC@OSV) VALUE(%SUBSTRING(&DCIDS 394 1))

/* Retrieve various attributes of this job                    */

     RTVJOBA    JOB(&DC@JOB) USER(&DC@USR) NBR(&DC@JNB) +
                OUTQ(&DC@OQN) OUTQLIB(&DC@OQL) +
                TYPE(&DC@JTP) USRLIBL(&DC@LBL) +
                SBMMSGQ(&DC@MQN) SBMMSGQLIB(&DC@MQL)
```

```
        IF          COND(&DC@JTP = '1') THEN(DO)
        CHGVAR      VAR(&DC@JTP) VALUE('I')
        ENDDO
        ELSE        CMD(DO)
        CHGVAR      VAR(&DC@JTP) VALUE('B')
        ENDDO

        CHGVAR      VAR(&DCEDS) VALUE(&LDAIDN)

/* Retrieve various attributes of this user                          */

      RTVUSRPRF  USRPRF(&DC@USR) JOBD(&JOBD) JOBDLIB(&JOBDLIB) +
                 GRPPRF(&DC@GRU) OWNER(&DC@GRO) +
                 GRPAUT(&DC@GRA) OUTQ(&OUTQ) OUTQLIB(&OUTQLIB)

        IF          COND(&JOBD = '*NONE') THEN(DO)
        CHGVAR      VAR(&JOBD) VALUE('QBATCH')
        CHGVAR      VAR(&JOBDLIB) VALUE('*LIBL')
        ENDDO

        IF          COND(&OUTQ = '*NONE') THEN(DO)
        CHGVAR      VAR(&OUTQ) VALUE('QPRINT')
        CHGVAR      VAR(&OUTQLIB) VALUE('*LIBL')
        ENDDO

        IF          COND(&OUTQ = '*WRKSTN') THEN(DO)
        CHGVAR      VAR(&OUTQ) VALUE('*JOBD')
        CHGVAR      VAR(&OUTQLIB) VALUE(' ')
        ENDDO

/* Set the number of user parameters to 0 in the EDS                 */

        CHGVAR      VAR(%SUBSTRING(&DCEDS 196 3)) VALUE('000')

/* Set the direct call indicator off                                 */

        CHGVAR      VAR(%SUBSTRING(&DCEDS 228 1)) VALUE('N')
```

```
/* Now set up the rest of the information data structure          */

/* Operating System                                               */
/*          If API QSZRTVPR is available, use it to get operating */
/*   system release level else use data areas and bypass QSZRTVPR. */

      CHKOBJ     OBJ(QSYS/QSZRTVPR) OBJTYPE(*PGM)
      MONMSG     MSGID(CPF9801) EXEC(DO)

      IF         COND(&DC@OSV = '2') THEN(RTVDTAARA +
                   DTAARA(QSYS/Q5738SS1 (1 6)) RTNVAR(&CPFLVL))
      ELSE       CMD(RTVDTAARA DTAARA(QSYS/Q5728SS1 (1 6)) +
                   RTNVAR(&CPFLVL))
      GOTO       CMDLBL(BYPASS)
      ENDDO

      CHGVAR     VAR(%BIN(&RSTLEN 1 4)) VALUE(22)
      CHGVAR     VAR(%BIN(&ERRCDE 1 4)) VALUE(0)
      CALL       PGM(QSYS/QSZRTVPR) PARM(&RESULT &RSTLEN +
                   &FORMAT &PRODUCT &ERRCDE)
      CHGVAR     VAR(&CPFLVL) VALUE(%SST(&RESULT 17 6))

BYPASS:
      IF         COND(&DC@OSV *GE '2') THEN(DO)
     CHGVAR %SUBSTRING(&DCIDS 501 1) %SUBSTRING(&CPFLVL 2 1)
     CHGVAR %SUBSTRING(&DCIDS 502 1) %SUBSTRING(&CPFLVL 4 1)
     CHGVAR %SUBSTRING(&DCIDS 503 1) %SUBSTRING(&CPFLVL 6 1)
     ENDDO
     ELSE DO
     CHGVAR %SUBSTRING(&DCIDS 501 2) %SUBSTRING(&CPFLVL 2 2)
     CHGVAR %SUBSTRING(&DCIDS 503 1) %SUBSTRING(&CPFLVL 6 1)
     ENDDO

/* Date                                                           */
      QSYS/RTVSYSVAL SYSVAL(QDATE) RTNVAR(&DC@DAT)
      QSYS/RTVSYSVAL SYSVAL(QDATFMT) RTNVAR(&DC@SDF)
```

```
 QSYS/RTVSYSVAL SYSVAL(QDATSEP) RTNVAR(&DC@SDS)
 QSYS/RTVSYSVAL SYSVAL(QTIMSEP) RTNVAR(&DC@STS)
CVTDAT      DATE(&DC@DAT) TOVAR(&DC@DMY) FROMFMT(*SYSVAL) +
            TOFMT(*DMY) TOSEP(*NONE)
CVTDAT      DATE(&DC@DAT) TOVAR(&DC@YMD) FROMFMT(*SYSVAL) +
            TOFMT(*YMD) TOSEP(*NONE)
CVTDAT      DATE(&DC@DAT) TOVAR(&DC@MDY) FROMFMT(*SYSVAL) +
            TOFMT(*MDY) TOSEP(*NONE)
CVTDAT      DATE(&DC@DAT) TOVAR(&DC@JUL) FROMFMT(*SYSVAL) +
            TOFMT(*JUL) TOSEP(*NONE)
CHGVAR %SUBSTRING(&DCIDS 504 32) +
(&DC@DAT||&DC@SDF||&DC@DMY||&DC@YMD||&DC@MDY||&DC@JUL)
CHGVAR VAR(%SST(&DCIDS 1011 2)) VALUE(&DC@SDS || &DC@STS)

 CHGVAR     VAR(%SUBSTRING(&DCIDS 970 9)) +
              VALUE('000000000')

 RTVSYSVAL  SYSVAL(QIGC) RTNVAR(&ML@IGM)
 CHGVAR     VAR(%SUBSTRING(&DCIDS 998 1)) VALUE(&ML@IGM)
 CHGVAR     VAR(%SUBSTRING(&DCIDS 999 1)) VALUE(' ')

CHGVAR %SUBSTRING(&DCIDS 536 342) +
(&DC@JTP||&DC@JOB||&DC@JNB||&DC@USR||&DC@OQN||&DC@OQL|| +
&DC@MQN||&DC@MQL||&DC@LBL)
 CHGVAR     VAR(%SUBSTRING(&DCIDS 878 10)) VALUE(&DC@GRU)
 CHGVAR     VAR(%SUBSTRING(&DCIDS 888 10)) VALUE(&DC@GRO)
 CHGVAR     VAR(%SUBSTRING(&DCIDS 898 10)) VALUE(&DC@GRA)

/*                                                          */
 CHGVAR     VAR(%SUBSTRING(&DCIDS 910 3)) VALUE(&UD@P#I)
 CHGVAR     VAR(%SUBSTRING(&DCIDS 1009 1)) VALUE('N')
 CHGVAR     VAR(%SUBSTRING(&DCEDS 1799 4)) VALUE(&UD@LNG)
 CHGVAR     VAR(%SUBSTRING(&DCIDS 909 1)) VALUE('D')

/* Insert default job submission details into DC@IDS area     */

 IF         COND(%SST(&JOBD 1 1) *NE '*') THEN(CHGVAR +
```

```
                VAR(%SST(&DCIDS 318 21)) VALUE(&JOBDLIB +
                  *TCAT '/' *CAT &JOBD))
        ELSE    CMD(CHGVAR VAR(%SUBSTRING(&DCIDS 318 21)) +
                  VALUE(&JOBD))

        CHGVAR  VAR(%SUBSTRING(&DCIDS 339 21)) VALUE('*JOBD')

        IF      COND(%SST(&OUTQ 1 1) *NE '*') THEN(CHGVAR +
                  VAR(%SST(&DCIDS 360 21)) VALUE(&OUTQLIB +
                  *TCAT '/' *CAT &OUTQ))
        ELSE    CMD(CHGVAR VAR(%SUBSTRING(&DCIDS 360 21)) +
                  VALUE(&OUTQ))

        CHGVAR  VAR(%SST(&DCEDS 487 1)) VALUE('Y')
        CHGVAR  VAR(%SST(&DCEDS 963 1)) VALUE(' ')

/* Now return control to the calling program DC@IDS                */
/*  has  been set up at this point.                                 */

        RETURN

/* If this point is reached the job ended abnormally .....         */

ERROR:
        CHGJOB  LOG(4 00 *SECLVL)
        MONMSG  MSGID(CPF0000 MCH0000)

ENDPGM:     ENDPGM
```

Program UD@CALL2

The source member for this program (UD@CALL2) can be found in file DC@F28 in the LANSA Data Library.

NOTE : Only the program source is supplied. The program object will need to be created before it can be executed.

```
H*%NATIVE
H*%AS  DLTPGM    PGM($$CRTLIB$$/UD@CALL2)
H*%AS  CRTRPGPGM PGM($$CRTLIB$$/UD@CALL2) SRCFILE(DCSRC) +
H*%AS             GENLVL(30) USRPRF(*OWNER) AUT(*USE) +
H*%AS             TGTRLS(*PRV)
H*%AS  CHGPGM    PGM($$CRTLIB$$/UD@CALL2) RMVOBS(*ALL)
H*====================================================================
H*
H*  Program      : UD@CALL2
H*  ~~~~~~~
H*  Date written : April, 1992
H*  ~~~~~~~~~~~~
H*  Author       : Ian Smith
H*  ~~~~~~
H*  Description  : This program is called to set up the DC@EDS
H*  ~~~~~~~~~~~    data structure for user controlled direct
H*                 invocation of LANSA functions.
H*
H*====================================================================
H*                 Program Amendment History
H*====================================================================
H*
H*  Ref    Date     Amendor Name   Brief Description of Amendment
H*  ---    ----     ------------   -----------------------------
H*  NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
H*  NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
H*  NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
H*
F*=============================================================================
F* File specifications
F*=============================================================================
FDC@F46V1IF  E           K        DISK                           UC
F                                                 KINFSR *PSSR
FDC@F60V1IF  E           K        DISK                           UC
F                                                 KINFSR *PSSR
E*=============================================================================
E* Array specifications
E*=============================================================================
E                    F60@D      30 10
E                    TXT     1   3 40
E                    PPP       550  1
E/COPY DC@F28,DC@ESPEC
E/COPY DC@F28,PR@ESPEC
I*=============================================================================
I* Data structure specifications
I*=============================================================================
I*
I* Partition information
I*
IF46IDS      DS
I                                          1 250 F46I01
I                                        251 500 F46I02
I                                        501 750 F46I03
I                                       7511000 F46I04
I                                        525 525 F46EDC
I                                        798 798 F46MLP
I                                        826 829 F46DLN
I*
I* Partition information (Mulitlingual)
I*
IF60IDS      DS                           1000
I                                          1  76 F60PMP
I                                         77 376 F60@D
I                                        377 436 F60@L
```

```
I                                          437 476 F60PHM
I                                          477 516 F60PRM
I                                          517 556 F60PEM
I                                          557 557 F60DEV
I                                          558 558 F60CNV
I                                          559 560 F60KCV
I                                          561 570 F60DCV
I                                          571 572 F60LCV
I                                          573 5740F60LDC
I*
IP#S         DS
I                                            1   5 PRO
I                                            6   6 SEP
I                                            7   9 P#I
I/COPY DC@F28,DC@ISPEC
I/COPY DC@F28,PR@ISPEC
I*============================================================================
I* Program information data structures
I*============================================================================
I*
I* Program information data structure (PIDS)
I* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
I*  $PGMNM : Indicates name of this program
I*
I$PIDS       SDS                             200
I                                          *PROGRAM $PGMNM
C*============================================================================
C* Parameter list specifications
C*============================================================================
C*
C* *ENTRY parameter list
C*
C           *ENTRY    PLIST
C                     PARM           DC@IDS
C                     PARM           DC@EDS
C*
C*============================================================================
```

```
C* Program mainline
C*=============================================================================
C*
C* If partition ID is blank, Change to default partition "SYS"
C*
C           DC@P#I    IFEQ *BLANKS
C                     MOVEL'SYS'     DC@P#I
C                     END
C*
C* Open partition file
C*
C                     OPEN DC@F46V1
C*
C* Attempt to locate the partition record
C*
C           DC@P#I    CHAINDC@R46V1             70
C*
C* Close the partition file
C*
C                     CLOSEDC@F46V1
C*
C* If not found, abort program
C*
C           *IN70     IFEQ '1'
C                     EXSR *PSSR
C                     END
C*
C* If language specified but this is not a multilingual partition
C* then abort the program
C*
C           ML@CLN    IFNE *BLANKS
C           F46MLP    ANDNE'Y'
C                     EXSR *PSSR
C                     END
C*
C* If this is a multilingual partition then also get the DC@F60
C* (multilingual partition details) record
```

```
C*
C           F46MLP    IFEQ 'Y'
C*
C* If language is blank & default language is also blank then set
C* the multilingual flag off.
C* Partition multilingual attributes have not been set up yet.
C*
C           ML@CLN    IFEQ *BLANKS
C           F46DLN    ANDEQ*BLANKS
C                     MOVEL'N'       F46MLP
C                     ELSE
C*
C* If language is blank, change to partition default language code
C*
C           ML@CLN    IFEQ *BLANKS
C                     MOVELF46DLN    ML@CLN
C                     END
C*
C* Open partition multilingual file
C*
C                     OPEN DC@F60V1
C*
C* Attempt to locate the partition language record
C*
C           F60K01    KLIST
C                     KFLD           DC@P#I
C                     KFLD           ML@CLN
C*
C           F60K01    CHAINDC@R60V1            70
C*
C* Close the partition multilingual file
C*
C                     CLOSEDC@F60V1
C*
C* If not found abort the program
C*
C           *IN70     IFEQ '1'
```

```
C                   EXSR *PSSR
C                   END
C*
C* If language is DBCS and system is not IGC capable,
C* then abort the program
C*
C           F60DBC  IFEQ 'Y'
C           ML@IGM  ANDNE'1'
C                   EXSR *PSSR
C                   END
C*
C                   END
C                   END
C*
C* Reset some details in the system definition block
C*
C                   MOVELF46P#I   DC@P#I
C                   MOVELF46P#D   DC@P#D
C                   MOVELF46MDL   DC@MDL
C                   MOVELF46SUI   DC@SUI
C                   MOVELF46DFN   DC@DFN
C                   MOVELF46EDC   DC@LDC
C                   MOVELDC@PRO   PRO
C                   MOVEL'/'      SEP
C                   MOVELDC@P#I   P#I
C                   MOVELP#S      DC@P#S
C*
C* Place all the SAA/CUA and extended details into the extended
C* area in DC@EDS (SAA/CUA and partition level details)
C*
C                   MOVELDC@P#I   DC@E#I
C                   MOVELF46I01   SA@I01
C                   MOVELF46I02   SA@I02
C                   MOVELF46I03   SA@I03
C                   MOVELF46I04   SA@I04
C                   MOVEL'Y'      ML@LRL
C                   MOVEL'N'      ML@IGL
```

```
C                     MOVEL'N'       ML@RLL
C*
C* If this is a multilingual partition then also set multilingual
C* details (overwriting some DC@F46 details)
C*
C           F46MLP    IFEQ 'Y'
C                     MOVELF60PLN    ML@CLN
C                     MOVELF60PLD    ML@CLD
C                     MOVELF60DBC    ML@IGL
C                     MOVELF60RLT    ML@RLL
C                     MOVELF60DEV    ML@DEV
C                     MOVELF60CNV    ML@CNV
C           ML@CNV    IFEQ 'Y'
C                     MOVELF60KCV    ML@KCV
C                     MOVELF60DCV    ML@DCV
C                     Z-ADDF60LDC    ML@LDC
C                     MOVELF60LCV    ML@LCV
C                     END
C           ML@IGL    IFEQ 'Y'
C           ML@RLL    OREQ 'Y'
C                     MOVEL'N'       ML@LRL
C                     ELSE
C                     MOVEL'Y'       ML@LRL
C                     END
C                     MOVELF60P#D    DC@P#D
C                     MOVELF60PMP    SA@PMP
C                     MOVEAF60@D     SA@D
C                     MOVELF60@L     CHSA@L
C                     MOVELF60PHM    P#@PHM
C                     MOVELF60PRM    P#@PRM
C                     MOVELF60PEM    P#@PEM
C                     MOVELF60MSG    ML@MSG
C                     END
C*
C* Set up SAA/CUA details of applicable
C*
C           F46SAA    IFEQ 'Y'
```

```
C                   MOVEL'Y'        SA@SAA
C                   MOVELSA@KEH     DC@CEX
C                   MOVELSA@KCN     DC@CMN
C                   MOVELSA@KMS     DC@CDM
C                   MOVELSA@KAD     DC@CAD
C                   MOVELSA@KCH     DC@CCH
C                   MOVELSA@KDL     DC@CDL
C                   MOVELSA@KPR     DC@CPM
C*
C* ELSE if CUA/SAA details do not apply
C*
C                   ELSE
C                   MOVEL'N'        SA@SAA
C                   Z-ADD0          SA@L
C                   MOVELDC@CEX     SA@KEH
C                   MOVEL'Exit      'SA@DEH
C                   Z-ADD4          SA@LEH
C                   MOVELDC@CEX     SA@KEL
C                   MOVEL'Exit      'SA@DEL
C                   Z-ADD4          SA@LEL
C                   MOVELDC@CMN     SA@KCN
C                   MOVEL'Cancel    'SA@DCN
C                   Z-ADD6          SA@LCN
C                   MOVELDC@CDM     SA@KMS
C                   MOVEL'Messages'SA@DMS
C                   Z-ADD8          SA@LMS
C                   MOVELDC@CAD     SA@KAD
C                   MOVEL'Add       'SA@DAD
C                   Z-ADD3          SA@LAD
C                   MOVELDC@CCH     SA@KCH
C                   MOVEL'Change    'SA@DCH
C                   Z-ADD6          SA@LCH
C                   MOVELDC@CDL     SA@KDL
C                   MOVEL'Delete    'SA@DDL
C                   Z-ADD6          SA@LDL
C                   MOVEL'23'       SA@KHC
C                   MOVEL'Contents'SA@DHC
```

```
C                   Z-ADD8          SA@LHC
C                   MOVEL'02'       SA@KHE
C                   MOVEL'Ex help 'SA@DHE
C                   Z-ADD7          SA@LHE
C                   MOVELDC@CPM     SA@KPR
C                   MOVEL'Prompt  'SA@DPR
C                   Z-ADD6          SA@LPR
C                   END
C*
C* Handle blank values in process/menu options
C*
C           P#@PHM  IFEQ *BLANKS
C                   MOVELTXT,01     P#@PHM
C                   END
C           P#@PRM  IFEQ *BLANKS
C                   MOVELTXT,02     P#@PRM
C                   END
C           P#@PEM  IFEQ *BLANKS
C                   MOVELTXT,03     P#@PEM
C                   END
C*
C* Set other defaults
C*
C           DC@HCH  IFEQ *BLANKS
C                   MOVEL'Ý'        DC@HCH
C                   END
C*
C           DC@VCH  IFEQ *BLANKS
C                   MOVEL'-'        DC@VCH
C                   END
C*
C           DC@CYY  IFEQ *BLANKS
C                   MOVEL'99'       DC@CYY
C                   END
C*
C           DC@CLE  IFEQ *BLANKS
C                   MOVEL'19'       DC@CLE
```

```
C                   END
C*
C           DC@CGT  IFEQ *BLANKS
C                   MOVEL'19'       DC@CGT
C                   END
C*
C           DC@DPC  IFNE '.'
C           DC@DPC  ANDNE','
C                   MOVEL'.'        DC@DPC
C                   END
C*
C           DC@OSV  IFEQ *BLANK
C           DC@CEE  IFEQ '3X'
C                   MOVEL'1'        DC@OSV          AS/400
C                   ELSE
C                   MOVEL'0'        DC@OSV          S/38
C                   END
C                   END
C*
C           DC@EOA  IFEQ *BLANKS
C                   MOVEL'MD'       DC@EOA
C                   END
C*
C* Set up additional variables
C*
C                   MOVE *BLANKS    DC@DEX
C                   MOVELSA@DEH     DC@DEX
C                   Z-ADDSA@LEH     DC@LEX
C                   MOVEL'E'        DC@TEX
C                   Z-ADD0          DC@NUP
C*
C* If required map the external list into the exchange list
C*
C           DC@EXL  IFEQ 'Y'
C                   MOVEL'$AC'      @3ACIN  3
C                   EXSR @3EXCH
C                   END
```

```
C*
C* SETON LR and return control to the caller
C*
C                   MOVE '1'        *INLR
C                   RETRN
C*
C*=========================================================
C*
C* @3EXCH : Set exchange list for 3GL CALL command
C*
C*=========================================================
C           @3EXCH  BEGSR
C*
C                   Z-ADD0          PR@NXL
C                   Z-ADD1          PR@NXP
C*
C                   CALL 'M@EXCHL'
C                   PARM            @3ACTN  3
C                   PARM            DC@IDS
C                   PARM            DC@EDS
C                   PARM            PR@IDS
C*
C                   ENDSR
C*=========================================================================
C*
C* *PSSR : Standard error trap / handling routine
C*
C*=========================================================================
C           *PSSR   BEGSR
C*
C                   ENDSR'*CANCL'
**
Display process or function HELP text
Return to
Exit from system
```

Examples of Use

Overview of Examples

All of the following examples are based on LANSA Release 5.0 at PC level F4. They have been written to execute on an AS/400 with a version of OS/400 at V1R3M0 or higher. The functions and processes used by these examples are a part of the shipped Demonstration Personnel System.

The examples shown below are provided with the LANSA Product. They are shipped initially with LANSA Release 5.0 PC Level F6. The source is stored in file DC@F28 in the LANSA Data Library with the following member names:

UD@FUNC1	Call a LANSA Function using RPG/400
UD@FUNC2	Call a LANSA Function using Control Language
UD@PROC1	Call a LANSA Process using RPG/400
UD@PROC2	Call a LANSA Process using Control Language

Executing the LANSA INQUIRE Function with RPG/400

The source member for this program (UD@FUNC1) can be found in file DC@F28 in the LANSA Data Library.

The following sample RPG/400 code is an example of directly calling a LANSA function. It shows the basics necessary to set up all relevent information for a function.

The program UD@FUNC1 calls the LANSA function INQUIRE - "Browse/Maintain Employee and Skill Files" and passes to the function, via the exchange list, a value for the field EMPNO - "Employee number".

This example executes the function in English.

NOTE : The employee number will only be exchanged if the INQUIRE function has been created with access to 3GL exchange list flag set on.

```
H*%NATIVE
H*%AS  DLTPGM    PGM(DC@TOOLLIB/UD@FUNC1)
H*%AS  CRTRPGPGM PGM(DC@TOOLLIB/UD@FUNC1) SRCFILE(DCSRC) +
H*%AS  GENLVL(30) USRPRF(*OWNER) AUT(*EXCLUDE) +
H*%AS  OPTION(*NOXREF) SRCMBR(G6UD@FUNC1)
H*====================================================================
H*                    Basic Program Details
H*====================================================================
H*
H*  Program Name   : UD@FUNC1
H*
H*  Date written   : 27th April 1992
H*
H*  Authors Name   : Ian Smith
H*
H*  Description    : This RPG program directly invokes the DEMO
H*                   function ENROL from PSLSYS in partition DEM.
H*                   The program is also passed (via exchange ) the
H*                   employee unique ID.
H*
H*====================================================================
H/EJECT
H*====================================================================
H*                  Program Amendment History
H*====================================================================
H*
H*  Ref    Date      Amendor Name    Brief Description of Amendment
H*  ---    ----      ------------    ------------------------------
H*  NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
H*  NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
H*  NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
H*
H*=================================================================
H/EJECT
F*=================================================================
F*                        File Specifications
F*=================================================================
F/EJECT
E*=================================================================
E*                        Array Specifications
E*=================================================================
E                    UD      1   2 40
I*=================================================================
I*                     Data Structure Specifications
I*=================================================================
I/COPY DC@F28,DC@ISPEC
I/COPY DC@F28,PR@ISPEC
I*
I* Program Information Data Structure
I*
I$PIDS      SDS                           200
I                                       *PROGRAM $PGMNM
C/EJECT
C*=================================================================
C*                       Program mainline
C*=================================================================
C*
C* Set up System Information Data Structure (DC@IDS)
C*
C* NOTE : This does not have to be performed for every process
C*        or function that is to be executed. It can be set up
C*        once and passed from program to program.
C*
C*
C                   MOVEL'DC@PGMLI'UD@PGL          Program Library
C                   MOVE 'B '      UD@PGL
C*
C                   CALL 'UD@CALL1'            99
```

```
C                   PARM            UD@PGL 10
C                   PARM 'DEM'      UD@P#I  3       Partition ID
C                   PARM 'ENG '     UD@LNG  4       Language Code
C                   PARM            DC@IDS
C                   PARM            DC@EDS
C*
C* Set up Extended System Information Data Structure (DC@EDS)
C*
C* NOTE : This does not have to be performed for every process
C*        or function that is to be executed. It can be set up
C*        once and passed from program to program.
C*
C                   CALL 'UD@CALL2'             99
C                   PARM            DC@IDS
C                   PARM            DC@EDS
C*
C* Set up the Process Information Data structure (PR@IDS)
C* Clear the exchange list
C*
C                   Z-ADD0          PR@NXL
C                   Z-ADD1          PR@NXP
C*
C                   MOVEL'INQUIRE'  PR@FUN          Function name
C                   MOVELUD,1       PR@FUD          Function Desc.
C                   MOVEL*BLANKS    PR@NXT
C                   MOVEL*BLANKS    PR@NXD
C                   MOVEL'MNU'      PR@TYP          Process type
MNU,FUN
C*                                                  CMD, HLP, or EXT
C                   MOVEL$PGMNM     PR@RMQ          Message queue
C                   MOVEL'PSLSYS'   PR@PRO          Process name
C                   MOVELUD,2       PR@PRD          Process
description
C                   MOVEL'L'        PR@RCL          Heavy = H, Light =
L
C                   MOVEL'DIS'      PR@MDE
C                   MOVEL*BLANKS    PR@VER
```

```
C                   MOVELSA@KEH     PR@CEX
C                   MOVELSA@KCN     PR@CMN
C                   MOVELSA@KMS     PR@CDM
C                   MOVELSA@KAD     PR@CAD
C                   MOVELSA@KCH     PR@CCH
C                   MOVELSA@KDL     PR@CDL
C                   MOVEL'00'       PR@CU1
C                   MOVEL'00'       PR@CU2
C                   MOVEL'00'       PR@CU3
C                   MOVEL'00'       PR@CU4
C                   MOVEL'00'       PR@CU5
C                   MOVE *BLANKS    PR@CD1
C                   MOVE *BLANKS    PR@CD2
C                   MOVE *BLANKS    PR@CD3
C                   MOVE *BLANKS    PR@CD4
C                   MOVE *BLANKS    PR@CD5
C                   MOVEL*BLANKS    PR@KEY
C                   MOVEL*BLANKS    PR@ELK
C*
C* $PGMNM is the name of this program
C* from the Program Information Data Structure (PIDS)
C*
C                   MOVEL$PGMNM     PR@RMQ
C                   MOVE '01'       PR@NIC
C                   MOVEL'Y'        PR@RQR
C                   MOVEL'Y'        PR@SAA
C                   MOVE 'N'        PR@EPR
C*
C* Set up the exchange list
C*
C* Set the employee number value
C*
C                   MOVEL'EMPNO'    UD@FLD 15
C                   MOVE 'A0050'    UD@FLD
C                   MOVEL'B1234'    UD@VAL  5
C*
C* Put the employee number on the list
```

```
C*
C                   CALL 'M@EXCHL'
C                   PARM 'PUT'     UD@ACT  3
C                   PARM           UD@FLD
C                   PARM           UD@VAL
C                   MOVEL'B1234'   UD@VAL  5
C*
C* Map external (3GL) exchange list into LANSA exchange list
C*
C                   CALL 'M@EXCHL'
C                   PARM '$AC'     UD@ACT  3
C                   PARM           DC@IDS
C                   PARM           DC@EDS
C                   PARM           PR@IDS
C*
C* Call the function INQUIRE
C*
C                   CALL '@INQUIRE'            99  Function name
C                   PARM           DC@IDS
C                   PARM           DC@EDS
C                   PARM           PR@IDS
C*
C* Map LANSA exchange list into external (3GL) exchange list
C*
C                   CALL 'M@EXCHL'
C                   PARM '$RC'     UD@ACT  3
C                   PARM           DC@IDS
C                   PARM           DC@EDS
C                   PARM           PR@IDS
C*
C* Get the employee number from the list
C*
C                   CALL 'M@EXCHL'
C                   PARM 'GET'     UD@ACT  3
C                   PARM           UD@FLD
C                   PARM           UD@VAL
C*
```

```
C                   MOVEL'1'        *INLR
C                   RETRN
C/EJECT
C*====================================================================
C* *PSSR : Handle / trap a total program failure.
C*         This routine when an unexpected failure of the program
occurs.
C*         This is done automatically by RPG/400 and is not under user
C*         control. Note that the entire program terminates when this
C*         routine is invoked.
C*====================================================================
C           *PSSR     BEGSR
C*
C* Set the return code to indicate a "bad" return has occured ....
C*
C                   MOVE 'N'        DC@RET
C*
C* Return control to the calling program (note that LR is on) ....
C*
C                   MOVE '1'        *INLR
C                   RETRN
C*
C* If this point is reached, abort program using the *CANCL option ...
C*
C                   ENDSR'*CANCL'
**
Browse/Maintain Employee and Skill Files
Personnel System Main Menu
```

Executing the LANSA INQUIRE Function with CL/400

The source member for this program (UD@FUNC2) can be found in file DC@F28 in the LANSA Data Library.

This example performs exactly the same processing as the previous sample code, except this program has been written using Control Language.

This example executes the function in English.

```
/*    %NATIVE                                                        */
/*    %AS  DLTPGM    PGM($$CRTLIB$$/UD@FUNC2)                       */
/*    %AS  CRTCLPGM  PGM($$CRTLIB$$/UD@FUNC2)   SRCFILE(DCSRC) +    */
/*    %AS            LOG(*NO) USRPRF(*OWNER) AUT(*EXCLUDE) +       */
/*    %AS            ALWRTVSRC(*NO)                                 */
/*    %AS  CHGPGM    PGM($$CRTLIB$$/UD@FUNC2)   RMVOBS(*ALL)        */
/*    %AS  CRTDUPOBJ OBJ(UD@FUNC2)   FROMLIB($$CRTLIB$$)  +         */
/*    %AS            OBJTYPE(*PGM) TOLIB($$WRKLIB$$)                */
/*===============================================================*/
/*               Basic Program Details                           */
/*===============================================================*/
/*                                                                 +
   Program Name  : UD@FUNC2                                        +
                                                                   +
   Date written  : 27th April 1992                                 +
                                                                   +
   Authors Name  : Ian Smith                                       +
                                                                   +
   Description   : This program directly calls a LANSA function    +
                   INQUIRE. It also exchanges an employee number to +
                   this function.                                  +
                                                                  */
/*===============================================================*/
/*               Program Amendment History                       */
/*===============================================================*/
/*                                                                 +
```

```
   Ref    Date     Amendor Name   Brief Description of Amendment      +
   ---    ----     ------------   ------------------------------      +
   NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  +
   NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  +
   NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  +
                                                                       */
/*====================================================================*/

UD@FUNC2:   PGM

/*====================================================================*/
/*                    Variable Declarations                           */
/*====================================================================*/

          DCL       VAR(&DC@IDS) TYPE(*CHAR) LEN(1024)
          DCL       VAR(&DC@EDS) TYPE(*CHAR) LEN(2500)
          DCL       VAR(&PR@IDS) TYPE(*CHAR) LEN(5000)

          DCL       VAR(&$PGMNM) TYPE(*CHAR) LEN(10)
          DCL       VAR(&DC@RET) TYPE(*CHAR) LEN(1)

          DCL       VAR(&UD@PGL) TYPE(*CHAR) LEN(10)
          DCL       VAR(&UD@P#I) TYPE(*CHAR) LEN(3)
          DCL       VAR(&UD@LNG) TYPE(*CHAR) LEN(4)

/*====================================================================*/
/*                    Global Error Handler                            */
/*====================================================================*/

          MONMSG    MSGID(CPF0000 MCH0000) EXEC(GOTO +
                      CMDLBL(ARGERR))

/*====================================================================*/
/*                     Program Mainline                               */
/*====================================================================*/

/* Set up program name                                                */
```

```
/*                                                                +
    NOTE : This does not have to be performed for every process +
           or function that is to be executed. It can be set up +
           once and passed from program to program.             +
                                                                 */

            CHGVAR      VAR(&$PGMNM) VALUE('UD@FUNC2')

/*  Set up the System information data structure (DC@IDS)         */

/*                                                                +
    NOTE : This does not have to be performed for every process +
           or function that is to be executed. It can be set up +
           once and passed from program to program.             +
                                                                 */

            CHGVAR      VAR(&UD@PGL) VALUE('DC@PGMLIB') /* Program +
                          library */
            CHGVAR      VAR(&UD@P#I) VALUE('DEM') /* Partition ID */
            CHGVAR      VAR(&UD@LNG) VALUE('ENG ') /* Language Code */
                        /* English Language */
            CALL        PGM(UD@CALL1) PARM(&UD@PGL &UD@P#I &UD@LNG +
                          &DC@IDS &DC@EDS)
            MONMSG      MSGID(CPF0000 MCH0000) EXEC(GOTO +
                          CMDLBL(ARGERR))

/*  Set up the Extended System information data structure (DC@EIDS)*/

            CALL        PGM(UD@CALL2) PARM(&DC@IDS &DC@EDS)
            MONMSG      MSGID(CPF0000 MCH0000) EXEC(GOTO +
                          CMDLBL(ARGERR))

/*  Set up the Process Information Data structure (PR@IDS)        */

/*  Clear the exchange list                                       */
```

```
CHGVAR    VAR(%SST(&PR@IDS 2501 3)) VALUE(X'00000F')
CHGVAR    VAR(%SST(&PR@IDS 2504 3)) VALUE(X'00001F')

CHGVAR    VAR(%SST(&PR@IDS 1 7)) VALUE('INQUIRE') /* +
            Function name */
CHGVAR    VAR(%SST(&PR@IDS 11 40)) +
            VALUE('Browse/Maintain Employee and +
            Skill files') /* Function Description */

CHGVAR    VAR(%SST(&PR@IDS 51 7)) VALUE(' ')
CHGVAR    VAR(%SST(&PR@IDS 61 40)) VALUE(' ')
CHGVAR    VAR(%SST(&PR@IDS 241 3)) VALUE('MNU') /* +
            Process Type (MNU, CMD, FUN, HLP, EXT) */

CHGVAR    VAR(%SST(&PR@IDS 244 10)) VALUE('PSLSYS') /* +
            Process name */
CHGVAR    VAR(%SST(&PR@IDS 254 40)) VALUE('          +
            Personnel System Main Menu') /* Process +
            Description */

CHGVAR    VAR(%SST(&PR@IDS 294 1)) VALUE('Y')
CHGVAR    VAR(%SST(&PR@IDS 295 3)) VALUE('DIS')
CHGVAR    VAR(%SST(&PR@IDS 298 3)) VALUE(' ')

CHGVAR    VAR(%SST(&PR@IDS 299 2)) VALUE(%SST(&DC@EDS +
            1110 2))
CHGVAR    VAR(%SST(&PR@IDS 301 2)) VALUE(%SST(&DC@EDS +
            1092 2))
CHGVAR    VAR(%SST(&PR@IDS 303 2)) VALUE(%SST(&DC@EDS +
            1102 2))
CHGVAR    VAR(%SST(&PR@IDS 305 2)) VALUE(%SST(&DC@EDS +
            1138 2))
CHGVAR    VAR(%SST(&PR@IDS 307 2)) VALUE(%SST(&DC@EDS +
            1140 2))
CHGVAR    VAR(%SST(&PR@IDS 309 2)) VALUE(%SST(&DC@EDS +
            1142 2))
```

```
            CHGVAR      VAR(%SST(&PR@IDS 101 2)) VALUE('00')
            CHGVAR      VAR(%SST(&PR@IDS 103 2)) VALUE('00')
            CHGVAR      VAR(%SST(&PR@IDS 105 2)) VALUE('00')
            CHGVAR      VAR(%SST(&PR@IDS 107 2)) VALUE('00')
            CHGVAR      VAR(%SST(&PR@IDS 109 2)) VALUE('00')

            CHGVAR      VAR(%SST(&PR@IDS 111 8)) VALUE(' ')
            CHGVAR      VAR(%SST(&PR@IDS 119 8)) VALUE(' ')
            CHGVAR      VAR(%SST(&PR@IDS 127 8)) VALUE(' ')
            CHGVAR      VAR(%SST(&PR@IDS 135 8)) VALUE(' ')
            CHGVAR      VAR(%SST(&PR@IDS 143 8)) VALUE(' ')
            CHGVAR      VAR(%SST(&PR@IDS 151 6)) VALUE(' ')

 /*    &$PGMNM is the name of this program                          */

            CHGVAR      VAR(%SST(&PR@IDS 341 10)) VALUE(&$PGMNM) /* +
                          Message Queue */
            CHGVAR      VAR(%SST(&PR@IDS 961 2)) VALUE('01')
            CHGVAR      VAR(%SST(&PR@IDS 963 1)) VALUE('Y')
            CHGVAR      VAR(%SST(&PR@IDS 966 1)) VALUE('Y')
            CHGVAR      VAR(%SST(&PR@IDS 161 1)) VALUE('N')

/* Put the employee number on the exchange list                     */

            CALL        PGM(M@EXCHL) +
                        PARM('PUT' 'EMPNO     A0050' 'B1234')

/* Map external (3GL) exchange list into LANSA exchange list        */

            CALL        PGM(M@EXCHL) PARM('$AC' &DC@IDS &DC@EDS +
                          &PR@IDS)

/* Call the function                                                */

            CALL        PGM(@INQUIRE) PARM(&DC@IDS &DC@EDS &PR@IDS)
            MONMSG      MSGID(CPF0000 MCH0000) EXEC(GOTO +
                          CMDLBL(ARGERR))
```

```
/* Map LANSA exchange list into external (3GL) exchange list        */

            CALL        PGM(M@EXCHL) PARM('$RC' &DC@IDS &DC@EDS +
                          &PR@IDS)

 /* End of program processing logic                                 */

 ENDPGM:    CHGVAR      VAR(%SST(&DC@EDS 38 1)) VALUE(&DC@RET)
            RETURN

/*================================================================*/
/* ARGERR : Handle a detected error in argument(s) passed to program*/
/*                                                                 */
/* NOTE:  The entire program terminates when this logic is invoked. */
/*================================================================*/
ARGERR:

/* Set up a "bad" return code ....                                 */

            CHGVAR      VAR(&DC@RET) VALUE('N')
            MONMSG      MSGID(CPF0000 MCH0000)

/* Return control to the calling program ....                      */

            GOTO        CMDLBL(ENDPGM)

/*================================================================*/

            ENDPGM
```

Executing the LANSA PSLSYS Process with RPG/400

The source member for this program (UD@PROC1) can be found in file DC@F28 in the LANSA Data Library.

The following sample RPG/400 code is an example of directly calling a LANSA Process. It shows the basics necessary to set up all relevent information to run a LANSA Process.

The program UD@PROC1 calls the LANSA Process PSLSYS - "Personell System".

This example executes the process in English.

```
H*%NATIVE
H*%AS  DLTPGM    PGM(DC@TOOLLIB/UD@PROC1)
H*%AS  CRTRPGPGM PGM(DC@TOOLLIB/UD@PROC1) SRCFILE(DCSRC) +
H*%AS  GENLVL(30) USRPRF(*OWNER) AUT(*EXCLUDE) OPTION(*NOXREF)
H*====================================================================
H*                    Basic Program Details
H*====================================================================
H*
H*  Program Name   : UD@PROC1
H*
H*  Date written   : 27th April 1992
H*
H*  Authors Name   : Ian Smith
H*
H*  Description    : Sample RPG program to execute LANSA process
H*                   PSLSYS.
H*
H*====================================================================
H/EJECT
H*====================================================================
H*                  Program Amendment History
H*====================================================================
H*
H*  Ref    Date      Amendor Name   Brief Description of Amendment
```

```
H*  ---    ----     ------------   ----------------------------
H*  NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
H*  NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
H*  NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
H*
H*====================================================================
H/EJECT
F*====================================================================
F*                        File Specifications
F*====================================================================
F/EJECT
E*====================================================================
E*                        Array Specifications
E*====================================================================
E                     PD      1   2 40
F/EJECT
I*====================================================================
I*                     Data Structure Specifications
I*====================================================================
I/COPY DC@F28,DC@ISPEC
I/COPY DC@F28,PR@ISPEC
I*
I* Program Information Data Structure
I*
I$PIDS       DS                           200
I                                      *PROGRAM $PGMNM
I*
I* Process Name Information
I*
IUD@PIN      DS
I                                        5  14 UD@PNM
C/EJECT
C*====================================================================
C*                       Program mainline
C*====================================================================
C*
C* Set up System Information Data Structure (DC@IDS)
```

```
C*
C                   MOVEL'DC@PGMLI'UD@PGL          Program Library
C                   MOVE 'B '      UD@PGL
C*
C                   CALL 'UD@CALL1'            99
C                   PARM           UD@PGL 10
C                   PARM 'DEM'     UD@P#I  3       Partition ID
C                   PARM 'ENG '    UD@LNG  4       Language Code
C                   PARM           DC@IDS
C                   PARM           DC@EDS
C*
C* Set up EXtended System Information Data Structure (DC@EDS)
C*
C                   CALL 'UD@CALL2'            99
C                   PARM           DC@IDS
C                   PARM           DC@EDS
C*
C* Set up the Process Information Data structure (PR@IDS)
C*
C* Clear the exchange list
C*
C                   Z-ADD0         PR@NXL
C                   Z-ADD1         PR@NXP
C*
C                   MOVEL*BLANKS   PR@FUN          Function name
C                   MOVEL*BLANKS   PR@FUD          Function Desc.
C                   MOVEL*BLANKS   PR@NXT
C                   MOVEL*BLANKS   PR@NXD
C                   MOVEL'MNU'     PR@TYP          Process type
MNU,FUN
C*                                                 CMD, HLP, or EXT
C                   MOVEL$PGMNM    PR@RMQ          Message queue
C                   MOVEL'PSLSYS'  PR@PRO          Process name
C                   MOVELPD,1      PR@PRD          Process
description
C                   MOVEL'Y'       PR@RCL          Heavy = N, LIGHT =
Y
```

```
C                   MOVEL'DIS'      PR@MDE
C                   MOVEL*BLANKS    PR@VER
C                   MOVELSA@KEH     PR@CEX
C                   MOVELSA@KCN     PR@CMN
C                   MOVELSA@KMS     PR@CDM
C                   MOVELSA@KAD     PR@CAD
C                   MOVELSA@KCH     PR@CCH
C                   MOVELSA@KDL     PR@CDL
C                   MOVEL'00'       PR@CU1
C                   MOVEL'00'       PR@CU2
C                   MOVEL'00'       PR@CU3
C                   MOVEL'00'       PR@CU4
C                   MOVEL'00'       PR@CU5
C                   MOVE *BLANKS    PR@CD1
C                   MOVE *BLANKS    PR@CD2
C                   MOVE *BLANKS    PR@CD3
C                   MOVE *BLANKS    PR@CD4
C                   MOVE *BLANKS    PR@CD5
C                   MOVEL*BLANKS    PR@KEY
C                   MOVEL*BLANKS    PR@ELK
C*
C* $PGMNM is the name of this program
C* from the Program Information Data Structure (PIDS)
C*
C                   MOVEL$PGMNM     PR@RMQ
C                   MOVE '01'       PR@NIC
C                   MOVEL'Y'        PR@RQR
C                   MOVEL'Y'        PR@SAA
C                   MOVE 'N'        PR@EPR
C                   MOVEL$PGMNM     PR@RMQ          Message queue
C*
C* Set the process name
C*
C* NOTE : The process object name may vary between sites.
C*        The process object name may vary between partitions.
c*
C                 MOVEL'P@D00001'UD@PRO 10        Process object
```

```
C                   MOVE '01'       UD@PRO          name
C                   MOVEL'PSLSYS'   UD@PNM          Process Name
C*
C                   CALL UD@PRO                 99
C                   PARM            DC@IDS
C                   PARM            DC@EDS
C                   PARM            UD@PIN
C                   PARM            PR@IDS
C                   PARM *BLANKS    UPM@DS  1       User parameter
C*
C                   MOVEL'1'        *INLR
C                   RETRN
C/EJECT
C*====================================================================
C* *PSSR : Handle / trap a total program failure.
C*         This routine when an unexpected failure of the program
C*         occurs. This is done automatically by RPG/400 and is not
C*         under user control. Note that the entire program terminates
C*         when this routine is invoked.
C*====================================================================
C           *PSSR     BEGSR
C*
C* Set the return code to indicate a "bad" return has occured ....
C*
C                   MOVE 'N'        DC@RET
C*
C* Return control to the calling program (note that LR is on) ....
C*
C                   MOVE '1'        *INLR
C                   RETRN
C*
C* If this point is reached, abort program using the *CANCL option ...
C*
C                   ENDSR'*CANCL'
**
Personnel System Main Menu
```

Executing the LANSA PSLSYS Process with CL/400

The source member for this program (UD@PROC2) can be found in file DC@F28 in the LANSA Data Library.

This example performs exactly the same processing as the previous sample code, except this program has been written using Control Language.

This example executes the process in French.

```
/*    %NATIVE                                                    */
/*    %AS  DLTPGM    PGM($$CRTLIB$$/UD@PROC2)                    */
/*    %AS  CRTCLPGM  PGM($$CRTLIB$$/UD@PROC2)   SRCFILE(DCSRC) + */
/*    %AS            LOG(*NO) USRPRF(*OWNER) AUT(*EXCLUDE) +     */
/*    %AS            ALWRTVSRC(*NO)                              */
/*    %AS  CHGPGM    PGM($$CRTLIB$$/UD@PROC2)   RMVOBS(*ALL)     */
/*    %AS  CRTDUPOBJ OBJ(UD@PROC2)   FROMLIB($$CRTLIB$$)  +      */
/*    %AS            OBJTYPE(*PGM) TOLIB($$WRKLIB$$)             */
/*============================================================*/
/*               Basic Program Details                        */
/*============================================================*/
/*                                                              +
   Program Name  : UD@PROC2                                     +
                                                                +
   Date written  : 27th April 1992                              +
                                                                +
   Authors Name  : Ian Smith                                    +
                                                                +
   Description   : Sample CL program to execute LANSA process PSLSYS  +
                   in French.                                   +
                                                               */
/*============================================================*/
/*               Program Amendment History                    */
/*============================================================*/
/*                                                              +
   Ref    Date     Amendor Name   Brief Description of Amendment  +
   ---    ----     ------------   -----------------------------  +
```

```
   NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  +
   NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  +
   NNNNN  DD/DD/DD XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  +
                                                                      */
/*================================================================*/

UD@PROC2:   PGM

/*================================================================*/
/*                    Variable Declarations                       */
/*================================================================*/

            DCL         VAR(&DC@IDS) TYPE(*CHAR) LEN(1024)
            DCL         VAR(&DC@EDS) TYPE(*CHAR) LEN(2500)
            DCL         VAR(&PR@IDS) TYPE(*CHAR) LEN(5000)

            DCL         VAR(&$PGMNM) TYPE(*CHAR) LEN(10)
            DCL         VAR(&DC@RET) TYPE(*CHAR) LEN(1)

            DCL         VAR(&UD@PGL) TYPE(*CHAR) LEN(10)
            DCL         VAR(&UD@P#I) TYPE(*CHAR) LEN(3)
            DCL         VAR(&UD@LNG) TYPE(*CHAR) LEN(4)
            DCL         VAR(&UD@PRO) TYPE(*CHAR) LEN(10)
            DCL         VAR(&UD@PNM) TYPE(*CHAR) LEN(10)
            DCL         VAR(&UD@PIN) TYPE(*CHAR) LEN(14)
            DCL         VAR(&UPM@DS) TYPE(*CHAR) LEN(1)

/*================================================================*/
/*                    Global Error Handler                        */
/*================================================================*/

            MONMSG      MSGID(CPF0000 MCH0000) EXEC(GOTO +
                          CMDLBL(ARGERR))
```

```
/*============================================================*/
/*                      Program Mainline                      */
/*============================================================*/

/* Set up program name                                        */

          CHGVAR     VAR(&$PGMNM) VALUE('UD@PROC2')

/*  Set up the System information data structure (DC@IDS)     */

          CHGVAR     VAR(&UD@PGL) VALUE('DC@PGMLIB') /* Program +
                       library */
          CHGVAR     VAR(&UD@P#I) VALUE('DEM') /* Partition ID */
          CHGVAR     VAR(&UD@LNG) VALUE('FRA ') /* Language Code */
                     /* French Language */
          CALL       PGM(UD@CALL1) PARM(&UD@PGL &UD@P#I &UD@LNG +
                       &DC@IDS &DC@EDS)
          MONMSG     MSGID(CPF0000 MCH0000) EXEC(GOTO +
                       CMDLBL(ARGERR))

/*  Set up the Extended System information data structure (DC@EIDS)*/

          CALL       PGM(UD@CALL2) PARM(&DC@IDS &DC@EDS)
          MONMSG     MSGID(CPF0000 MCH0000) EXEC(GOTO +
                       CMDLBL(ARGERR))

/*  Set up the Process Information Data structure (PR@IDS)    */

/*  Clear the exchange list                                   */

          CHGVAR     VAR(%SST(&PR@IDS 2501 3)) VALUE(X'00000F')
          CHGVAR     VAR(%SST(&PR@IDS 2504 3)) VALUE(X'00001F')

          CHGVAR     VAR(%SST(&PR@IDS 1 7)) VALUE(' ') /* +
                       Function name */
          CHGVAR     VAR(%SST(&PR@IDS 11 40)) VALUE(' ') /* +
                       Function Description */
```

```
CHGVAR      VAR(%SST(&PR@IDS 51 7)) VALUE(' ')
CHGVAR      VAR(%SST(&PR@IDS 61 40)) VALUE(' ')
CHGVAR      VAR(%SST(&PR@IDS 241 3)) VALUE('MNU') /* +
              Process Type (MNU, CMD, FUN, HLP, EXT) */

CHGVAR      VAR(%SST(&PR@IDS 244 10)) VALUE('PSLSYS') /* +
              Process name */
CHGVAR      VAR(%SST(&PR@IDS 254 40)) VALUE('         +
              Menu principal de gestion du personnel') +
              /* Process Description */

CHGVAR      VAR(%SST(&PR@IDS 294 1)) VALUE('Y')
CHGVAR      VAR(%SST(&PR@IDS 295 3)) VALUE('DIS')
CHGVAR      VAR(%SST(&PR@IDS 298 3)) VALUE(' ')

CHGVAR      VAR(%SST(&PR@IDS 299 2)) VALUE(%SST(&DC@EDS +
              1110 2))
CHGVAR      VAR(%SST(&PR@IDS 301 2)) VALUE(%SST(&DC@EDS +
              1092 2))
CHGVAR      VAR(%SST(&PR@IDS 303 2)) VALUE(%SST(&DC@EDS +
              1102 2))
CHGVAR      VAR(%SST(&PR@IDS 305 2)) VALUE(%SST(&DC@EDS +
              1138 2))
CHGVAR      VAR(%SST(&PR@IDS 307 2)) VALUE(%SST(&DC@EDS +
              1140 2))
CHGVAR      VAR(%SST(&PR@IDS 309 2)) VALUE(%SST(&DC@EDS +
              1142 2))

CHGVAR      VAR(%SST(&PR@IDS 101 2)) VALUE('00')
CHGVAR      VAR(%SST(&PR@IDS 103 2)) VALUE('00')
CHGVAR      VAR(%SST(&PR@IDS 105 2)) VALUE('00')
CHGVAR      VAR(%SST(&PR@IDS 107 2)) VALUE('00')
CHGVAR      VAR(%SST(&PR@IDS 109 2)) VALUE('00')

CHGVAR      VAR(%SST(&PR@IDS 111 8)) VALUE(' ')
CHGVAR      VAR(%SST(&PR@IDS 119 8)) VALUE(' ')
```

```
            CHGVAR      VAR(%SST(&PR@IDS 127 8)) VALUE(' ')
            CHGVAR      VAR(%SST(&PR@IDS 135 8)) VALUE(' ')
            CHGVAR      VAR(%SST(&PR@IDS 143 8)) VALUE(' ')
            CHGVAR      VAR(%SST(&PR@IDS 151 6)) VALUE(' ')

 /*    &$PGMNM is the name of this program                          */

            CHGVAR      VAR(%SST(&PR@IDS 341 10)) VALUE(&$PGMNM) /* +
                          Message Queue */
            CHGVAR      VAR(%SST(&PR@IDS 961 2)) VALUE('01')
            CHGVAR      VAR(%SST(&PR@IDS 963 1)) VALUE('Y')
            CHGVAR      VAR(%SST(&PR@IDS 966 1)) VALUE('Y')
            CHGVAR      VAR(%SST(&PR@IDS 161 1)) VALUE('N')

 /* Set the process name                                           */
 /*                                                               +
       NOTE : The process object name may vary between sites.    +
              The process object name may vary between partitions.+
                                                                 */

            CHGVAR      VAR(&UD@PRO) VALUE('P@D0000101') /* Process +
                          Object name */
            CHGVAR      VAR(&UD@PNM) VALUE('PSLSYS') /* Process +
                          name */

            CHGVAR      VAR(%SST(&UD@PIN 5 10)) VALUE(&UD@PNM) /* +
                          Process Information */

/* Call the Process                                               */

            CALL        PGM(&UD@PRO) PARM(&DC@IDS &DC@EDS &UD@PIN +
                          &PR@IDS &UPM@DS)
            MONMSG      MSGID(CPF0000 MCH0000) EXEC(GOTO +
                          CMDLBL(ARGERR))

 /* End of program processing logic                               */
```

```
ENDPGM:     CHGVAR      VAR(%SST(&DC@EDS 38 1)) VALUE(&DC@RET)
            RETURN

/*================================================================*/
/* ARGERR : Handle a detected error in argument(s) passed to program*/
/*                                                                */
/* NOTE:  The entire program terminates when this logic is invoked. */
/*================================================================*/
ARGERR:

/* Set up a "bad" return code ....                                */

            CHGVAR      VAR(&DC@RET) VALUE('N')
            MONMSG      MSGID(CPF0000 MCH0000)

/* Return control to the calling program ....                     */

            GOTO        CMDLBL(ENDPGM)

/*================================================================*/

            ENDPGM
```

Index

There are no index entries for this guide.

Tell us what you think of this guide

To: **LANSA Development Manager**
Fax: **+61 (2) 9957 2657**
Email: ***lansamarketing@aspect.com.au***

LANSA/AD Open System Document
Internal Data Base and System Utilities

We hope you found this guide useful and informative. If you like what we've done, please let us know, if not, please tell us why, so that can make the guide better.

✓	Yes	No	No opinion
Does this guide meet your needs?			
Have you found the information accurate?			
Do you find the contents well organised?			
Is it easy to understand?			

What do you think we should do to improve this guide?

_____

_____

_____

_____

_____

_____

_____

Your
Name:_____

Company:_____

Tel: _____ Fax: _____

Email: _____

Thank you for taking the time to fill out this response.

Fold Under

If you would rather send your reponse by post,
or have your local LANSA distributor send it for you,
please fold on the dotted line and staple where indicated.

LANSA Development Manager
Aspect Computing Pty Ltd
Level 11,
122 Arthur Street
North Sydney
Australia
2060

Back - fold over