Astradyne (UK) Ltd
4 Knoll Road, Fleet, Hampshire. GU51 4PR

Tel: 01252 400955 Fax:01252 400956
Email: info@astradyne-uk.com

Technical Document

Document No: 07-02002

Document Title: Accessing the Local Data Area from an RPG Program

Category: Hints, Tips & FAQ

Functional Area: Programming

OS/400 Release:

Document Description:

A recent question asked through our support feature was "how can you access the Local Data Area (LDA) for a job and/or user?" Before answering that question, though, there are some points to note about the LDA:

1. The LDA is associated with a job and not the user. Each user can have multiple jobs running on the system and each job will have its own LDA.
2. As the name suggests, the LDA is "local" and therefore not accessible directly by other jobs on the system.
3. When a job is submitted to batch the new jobs LDA will be a copy of the LDA of the job that submitted it.

In order to use the jobs LDA within an RPG program we have to do a number of things. The first is to define an internal data structure to store the contents of the LDA within your program:

```
d LDA              DS
d  LDAData                 1   1024
d  Field1                  1     40
d  Field2                 41     50
d  Field3                 51    300
```

Next you have to link the external LDA to the data structure you have just defined. This only has to be done once, and should be – for example – in your initialisation subroutine:

```
c      *Dtaara        Define     *LDA           LDA
```

Finally, you use the **In** operation to read the contents of the LDA into your data structure and the **Out** operation to set the LDA with the contents of your data structure

```
c                     In         LDA
```

```
c                     Out        LDA
```

If you want to be able to read and process the LDA's for other jobs then you need to use a service program to perform the reads and writes locally, and to store the contents of each jobs LDA in a work file. This is obviously slightly more resource intensive than only accessing the current jobs LDA, but is a work around if needed.

The following service program can be used and bound to any programs making use of the LDA:

```
     H NoMain Option(*NoDebugIO)
     ******************************************************************
     **                                                              **
     **             A S T R A D Y N E   ( U K )   L T D              **
     **                                                              **
     **               4 Knoll Road, Fleet, Hampshire,                **
     **                      England. GU51 4PR                       **
     **                                                              **
     **           Tel: 01252 400955     Fax: 01252 400956            **
     **                E-mail: info@astradyne-uk.com                 **
     **                                                              **
     ******************************************************************
     **                                                              **
     **               S Y S T E M   U T I L I T I E S                **
     **                                                              **
     ******************************************************************
     **                (C) Astradyne (UK) Ltd, 2004.                 **
     **                                                              **
     **     This material is proprietary to Astradyne (UK) Ltd.      **
     **     All rights reserved.  The methods and techniques         **
     **     described herein are considered trade secrets and/or     **
     **     confidential.  Reproduction or distribution, in whole    **
     **     or in part, is forbidden except by express written       **
     **     permission of Astradyne (UK) Ltd.                        **
     **                                                              **
     ******************************************************************
     **                                                              **
     **                     PROGRAM DESCRIPTION€                    **
     **                                                              **
     **  Program    : LDASRVPGM                                      **
     **                                                              **
     **  Written By : Jonathan Mason              Date: 27-Sep-2004  **
     **                                                              **
     **  Description: LDA handling routines to enable other jobs to  **
     **               access/modify each others LDA.                 **
     **                                                              **
     **  Procedures : RtvJobLDA    Retrieve LDA for specified job     **
     **               SetJobLDA    Set LDA for specified job          **
     **               InzJobLDA    Initialise LDA for current job     **
     **                                                              **
     ******************************************************************
     FLDADATA   UF A E           K Disk
      * Local Data Area File

      ************************************
      * D A T A   D E F I N I T I O N S *
      ************************************

      * Program Data Structure
     D PgmDS          SDS
     D  JobName              244    253
     D  JobUser              254    263
     D  JobNumber            264    269
```

```
     * LDA Data Structure
    D JobLDA          DS
    D  JobLDAData             1   1024

     * Work Fields

     *********************************************
     * P R O T O T Y P E   D E F I N I T I O N S *
     *********************************************

     * Retrieve Job LDA
    d RtvJobLDA       pr          1024
    d  pJobName                     10    Value
    d  pJobUser                     10    Options(*NoPass)
    d  pJobNumber                    6    Options(*NoPass)

     * Set Job LDA
    d SetJobLDA       pr
    d  pJobLDA                    1024
    d  pJobName                     10    Value
    d  pJobUser                     10    Options(*NoPass)
    d  pJobNumber                    6    Options(*NoPass)

     * Initialise Job LDA
    d InzJobLDA       pr

     * Display Job LDA
    d DspJobLDA       pr
    d  pJobName                     10    Value
    d  pJobUser                     10    Options(*NoPass)
    d  pJobNumber                    6    Options(*NoPass)

     * Clear Job LDA
    d ClrJobLDA       pr
    d  pJobName                     10    Value
    d  pJobUser                     10    Options(*NoPass)
    d  pJobNumber                    6    Options(*NoPass)

     ************************************
     * G L O B A L   K E Y   L I S T S *
     ************************************

    c     KeyLDA        Klist
    c                   Kfld                    pJobName         10
    c                   Kfld                    pJobUser         10
    c                   Kfld                    pJobNumber        6

    c     *Dtaara       Define    *LDA          JobLDA

     ************************
     * P R O C E D U R E S *
     ************************

     * Retrieve Job LDA
    p RtvJobLDA       b                    Export

    d RtvJobLDA       pi          1024
    d  pJobName                     10    Value
    d  pJobUser                     10    Options(*NoPass)
    d  pJobNumber                    6    Options(*NoPass)

     * If the Job Name was passed as "*" or "*CURRENT" then
     * use the values from the SDS...
    c                   If        pJobName = '*' or pJobName = '*CURRENT'
    c                   Eval      pJobName = JobName
    c                   Eval      pJobUser = JobUser
    c                   Eval      pJobNumber = JobNumber
```

```
c                   EndIf

 * If the current job's LDA is being retrieved then retrieve it
 * directly rather than from the file.  This way we prevent other
 * jobs from modifying our LDA and affecting the running of our
 * program.  If you want to be able to modify another jobs LDA
 * then comment out the following If...Else...End apart from the
 * chain and record found conditions.  Note, however, if you do
 * this then you must use the InzJobLDA to set the contents in
 * the tracking file...
c                   If        pJobName = JobName and
c                             pJobUser = JobUser and
c                             pJobNumber = JobNumber
c                   In        JobLDA
c                   Return    JobLDA
c                   Else
c     KeyLDA        Chain(e)  LDAData
c                   If        %Found
c                   Return    PFJOBN
c                   Else
c                   Return    '*NOTFOUND'
c                   EndIf

c                   EndIf

p RtvJobLDA       e
 * ------------------------------------------------------------------------

 * Set Job LDA
p SetJobLDA       b

d SetJobLDA       pi
d  pJobLDA                    1024
d  pJobName                     10    Value
d  pJobUser                     10    Options(*NoPass)
d  pJobNumber                    6    Options(*NoPass)

 * If the Job Name was passed as "*" or "*CURRENT" then
 * use the values from the SDS...
c                   If        pJobName = '*' or pJobName = '*CURRENT'
c                   Eval      pJobName = JobName
c                   Eval      pJobUser = JobUser
c                   Eval      pJobNumber = JobNumber
c                   EndIf

 * We only set the current job's LDA, this way we prevent other
 * jobs from modifying our LDA and affecting the running of our
 * program.  If you want to be able to modify another jobs LDA
 * then move the lastšEndIf,so that it is just after thešOut,
 * operation...
c                   If        pJobName = JobName and
c                             pJobUser = JobUser and
c                             pJobNumber = JobNumber
c                   Eval      JobLDAData = pJobLDA
c                   Out       JobLDA

c     KeyLDA        Chain(e)  LDAData
c                   If        %Found
c                   Eval      PFLDA = pJobLDA
c                   Update    LDADataR
c                   Else
c                   Eval      PFJOBN = pJobName
c                   Eval      PFJOBU = pJobUser
c                   Eval      PFJOB# = pJobNumber
c                   Eval      PFJOBT = ' '
c                   Eval      PFLDA  = pJobLDA
c                   Write     LDADataR
```

```
c                   EndIf

c                   EndIf
c                   Return

p SetJobLDA       e
 * ------------------------------------------------------------------------
 * Initialise Job LDA
p InzJobLDA       b

d InzJobLDA       pi

 * Retrieve the current jobs LDA...
c                   In        JobLDA

 * ...and then set/store it...
c                   Callp     SetJobLDA (JobLDAData : '*')
c
c                   Return

p InzJobLDA       e
 * ------------------------------------------------------------------------
```

The source for the data file used by the service program is shown below:

```
A          R LDADATAR
**
A            PFJOBN        10          TEXT('Job Name')
A            PFJOBU        10          TEXT('Job User')
A            PFJOB#         6          TEXT('Job Number')
A            PFJOBT         1          TEXT('Job Type')
A            PFLDA       1024          TEXT('LDA Contents')
**
A          K PFJOBN
A          K PFJOBU
A          K PFJOB#
```

The service program provides three sub-procedures that work with the LDA; RtvJobLDA, SetJobLDA and InzJobLDA.

RtvJobLDA (Retrieve Job LDA) is passed the job name, job user and job number of the job whose LDA is to be accessed. The job name parameter is the only mandatory one, and an asterisk ('*') can be passed if you are interested in the current job. If you are working with the current jobs LDA then this is retrieved directly using the **In** operation code and returned to the calling instruction. If you are trying to access the LDA for a different job then the job name, user and number are used to chain to the LDADATA file and the contents (if any) returned. Needless to say, the validity of the other jobs LDA value returned is dependent on these sub-procedures being used for all LDA access.

The SetJobLDA (Set Job LDA) sub-procedure is passed the contents of a 1024 character data structure as the value to be written to the LDA in addition to the job name, user and number. Similarly to the RtvJobLDA sub-procedure an asterisk can be passed to signify the current job. The procedure allows you to set the LDA for the current job only, using the **Out** operation code to update the actual LDA data area as well as the LDADATA file.

If you want to be able to update LDA's for other jobs then you should use the following alternative RtvJobLDA and SetJobLDA sub-procedures:

```
 * Retrieve Job LDA
```

```
p RtvJobLDA       b                    Export

d RtvJobLDA       pi          1024
d  pJobName                     10    Value
d  pJobUser                     10    Options(*NoPass)
d  pJobNumber                    6    Options(*NoPass)

 * If the Job Name was passed as "*" or "*CURRENT" then
 * use the values from the SDS...
c                   If        pJobName = '*' or pJobName = '*CURRENT'
c                   Eval      pJobName = JobName
c                   Eval      pJobUser = JobUser
c                   Eval      pJobNumber = JobNumber
c                   EndIf

 * Read the contents of the LDA from the LDADATA file...
c     KeyLDA        Chain(e)  LDAData
c                   If        %Found
c                   Return    PFJOBN
c                   Else
c                   Return    '*NOTFOUND'
c                   EndIf

p RtvJobLDA       e
 * ------------------------------------------------------------------------

 * Set Job LDA
p SetJobLDA       b

d SetJobLDA       pi
d  pJobLDA                  1024
d  pJobName                     10    Value
d  pJobUser                     10    Options(*NoPass)
d  pJobNumber                    6    Options(*NoPass)

 * If the Job Name was passed as "*" or "*CURRENT" then
 * use the values from the SDS...
c                   If        pJobName = '*' or pJobName = '*CURRENT'
c                   Eval      pJobName = JobName
c                   Eval      pJobUser = JobUser
c                   Eval      pJobNumber = JobNumber
c                   EndIf

 * Set the physical LDA for the current job and update the LDADATA
 * file for all jobs...
c                   If        pJobName = JobName and
c                             pJobUser = JobUser and
c                             pJobNumber = JobNumber
c                   Eval      JobLDAData = pJobLDA
c                   Out       JobLDA
c                   EndIf

c     KeyLDA        Chain(e)  LDAData
c                   If        %Found
c                   Eval      PFLDA = pJobLDA
c                   Update    LDADataR
c                   Else
c                   Eval      PFJOBN = pJobName
c                   Eval      PFJOBU = pJobUser
c                   Eval      PFJOB# = pJobNumber
c                   Eval      PFJOBT = ' '
c                   Eval      PFLDA  = pJobLDA
c                   Write     LDADataR
c                   EndIf

c                   Return
```

```
      p SetJobLDA        e
```

The InzJobLDA (Initialise Job LDA) sub-procedure carries no parameters and is used solely to copy the contents of the current jobs LDA to the LDADATA file. This should be called at the start of any program that uses the LDA.

** End of Document **