Considerations for Migration for ASNA Monarch® 3.1

Contents

Monarch Overview **1**

RPG Programs **1**
Op-Codes 1
Features 2
Program Status DS and File Information DS 3

CL Programs **4**
Commands Supported/Unsupported 5

Display Files **6**
CssClass Property Migration Considerations 6

Printer Files **7**

Database Files **7**
Dealing with application using multi-member files 7
Logical Field Restrictions 8
Unlocking Records 8

Appendix I - DDS Display File Keywords **9**
Display Field-Level Keywords 9
Display Record-Level Keywords 10
Display File-Level Keywords 11
Subfile Keywords 11
Display Attributes for DSPATR Keyword 12

Appendix II - DDS Print File Keywords **13**
Printer Field-Level Keywords 13
Printer Record-Level Keywords 14
Printer File-Level Keywords 14

Considerations for Migration Guide

Monarch Overview

ASNA Monarch® is a set of technologies that assist in the migration of RPG applications to the .NET platform. While it strives to provide seamless flow of the applications objects to the new platform, there are several considerations that must be undertaken and instances where complete like-to-like transformation is not possible.

This document provides information on the level of transparency and support achieved by Monarch 3.1 for the different object types. Strategies are suggested for dealing with those issues where manual intervention is required.

Monarch will continue to evolve as market requirements change; therefore, the information provided in this document is subject to change.

RPG Programs

Op-Codes

As of version 3.1, all RPG op codes are supported with the exception of those shown in the table below.

Unsupported
ACQ
ALLOC
DEALLOC
DEBUG
DSPLY
DUMP
FORCE
NEXT
POST
REALLOC
REL
RESET
SHTDN

Features

Overlapping Fields

The overlapping data structure field feature of RPG is used to remap areas of memory that allows the program to give multiple meanings to the same bytes. .NET strictly enforces type safety and abhors this kind of arbitrary reinterpretation of memory. Remapping is sometimes used in a very conflicting manner, but there are certain common usages that are convenient and valid.

We have identified two common idioms involving overlapping data fields: giving individual field names to elements of an array, and masquerading a data structure as a large character field; typically stored as an unformatted record or as a data area. AVR provides the commands *DclAliasGroup* and *DSLoad*/*DSDump* to facilitate these types of operations.

There is another common usage of remapping, extensively used across the RPG community which requires special handling. It involves partitioning a text field into subfields. The most prevalent and probably perverse use of this style involves dealing with dates. The Date (and Time) type did not appear in RPG until the mid nineties with the advent of RPG ILE. Prior to that, RPG programmers were forced to use a Character (or Decimal) type to hold a date field, and then subdivide the field into the year, month and day subfields. Another case of subfields deals with encoded data, for example composing an account number based on the division, department and project of the account. Monarch makes use of the new AVR command **DclAlias** and the **Overlay** keyword of **DclDsFld** to implement overlapping fields.

Overlapping data structure fields which are not wholly contained in a parent field are flagged as errors that have to be dealt with by hand. The technique to be used in these cases is the establishment of a property where any necessary concatenation and sub-stringing is done to produce the desired subfield.

Internally Described Files and EXCEPT

Monarch supports program-described printer files and its corresponding EXCEPTs. EXCEPTs used to update individual fields of database files are supported.

Monarch extends partial support to Database program described files. The Migrator must obtain properly described files and a product like ASNA ProStart can be used to create database files that accurately define their field composition.

As part of the automatic migration process, a future version of Monarch will be able to match up the external field names and types with the one found in the program I and O specs. Monarch will then emit appropriate renames or aliases to have the program refer accurately to the external file definition.

There are a few cases when this match up cannot be accomplished. These include:

- Multiple record formats within a single file.
- Overlapping fields that are not wholly contained in a parent field.
- Two renamed-fields in different files sharing the same name.
- Unmatched field name list between the I and the O specs.

Hexadecimal Constants

Constants of the form X'xxxx' are invalid, however, the H'xxxx' are valid.

Cycle

No current plans are in place to support the Cycle.

There are two possible uses of *"The Cycle"* in an OS/400 application:

- ***Heavy use of "The Cycle"*** - If an application relies too much on The Cycle, then it would be too complex for Monarch to provide a good translation for that code. The portion of the code that relies on the Cycle will have to be re-written manually using AVR. Matching records is an example of heavy use.
- ***Light use of "The Cycle"*** - If an application does not rely extensively on the cycle, the Migrator should translate those few lines manually using AVR code.

Embedded SQL

Embedded SQL will be supported in a future release. Currently any embedded SQL has to be translated by hand into ADO. See **How to Migrate Embedded SQL** in the Migrating Application User's Guide for more detailed information.

ILE RPG

Monarch 3.1 supports single and multi-module ILE RPG programs and Service programs. Procedures are also supported.

Program Status DS and File Information DS

A program-status data structure provides program exception/error information to the program. The PSDS is defined in the main source section; therefore, there is only one PSDS per module.

The location of the subfields in the PSDS is defined by special keywords or by predefined From and To positions. In order to access the subfields, you assign a name to each subfield. The following fields are supported on the Program Status DS:

Offset/Keyword	Description
1	Program
191	Job Start Date
244	Job
254	User
264	Job Number
270	Job Start Time
276	Program Start Date
282	Program Start Time
358	Current User
*PROC	Program
*PROGRAM	Program

The following fields are supported on the File Information DS:

Offset/Keyword	Description
261	Record
367	Flags
369	AID Byte
370	Cursor Row
371	Cursor Column
376	Relative Record Number
*FILE	File
*RECORD	Record Name

CL Programs

A CL Program has the capability of using over a thousand commands provided by OS/400, plus any number of user-created commands. Usage of CL can be divided into two major categories: **System Administration** and **Application Coordination**.

The administration of the **system** involves operations like the creation of user profiles and the save/restore procedures. These activities are not considered part of the application itself and Monarch does not attempt to facilitate such activities. Normal Windows techniques should be employed to affect these kinds of activities.

On its other personality, CL is used to set up the environment for RPG programs to run. Typical functions done by CL in this context are:

- Set up the Library List
- Override database file
- Select a different File or Member to be used by the RPG 'F' spec
- Establish a Query File
- Maintain application parameters in data area
- Allocate objects
- Control the Program Message Queue

These actions affect the OS/400 Job where programs are running and have an effect on all programs on the same job. Monarch Framework provides these facilities through a set of classes that supplement the .NET Framework.

Commands Supported/Unsupported

As of 3.1, Monarch **does** support the following list of commands:

- ADDLIBLE
- CALL/CALLB
- CHGVAR
- CHGDTAARA
- CHKOBJ for ObjType *FILE only
- CLRPFM
- CRTDUPOBJ for files only
- DCL / DCLF
- DLTF FILE (library/file name) only
- DO
- DLTOVR
- ELSE
- ENDDO
- ENDPGM
- GOTO
- IF
- INZPFM
- MONMSG
- OVRDBF/ OVRDSPF
- PGM
- RCVF
- RETURN
- RTVDTAARA
- RMVLIBLE
- RMVMSG
- RTVJOBA
- SNDF
- SNDRCVF

Display Files

Display files are migrated into Active Server Pages for .NET forms. The migration strategy for each display file is to create an ASP.NET form composed of two files. First, there is an .aspx file containing HTML, ASP, and Monarch elements describing the layout and data composition of the display using server controls provided by Monarch. Second, an .aspx.vr file contains the code-behind class which extends the ASNA.Monarch.WebDspF.Page class. This class is part of a framework providing the runtime support to make the display file appear to the user.

The Display File Agent creates the ASP.NET form taking as input the OS/400 display files DDS specifications. Most of the elements found in the DDS specification are migrated. However, there are two general features which today are not dealt with: Help and window widgets. The help system defined in the DDS is alien to the web model and it is not migrated. In the mid 90s, DDS incorporated a set of keywords to facilitate the creation of Windows-looking screens. Because specialized hardware was needed to render these keywords properly, the practice of using them never caught on. The one big exception is the support for the WINDOW keyword which is converted to a browser pop-up window.

Special Note: *During Migration, a record format name from the Legacy Source Display File will have an "_" (underscore) prefixed to it for the* ***ID*** *property when the .aspx file is created. This alters the record format name that might otherwise conflict with a keyword in the forms designer. The* ***Alias*** *property is set to the original record format name used by the compiler.*

For example, if the DDS record format name is "MYWINDOW", the .aspx file would contain ***id="_MYWINDOW"*** *and* ***Alias="MYWINDOW"****. The same applies to subfile and subfile control record format names.*

CssClass Property Migration Considerations

During migration, the Solution Builder in Monarch Cocoon updates the CssClass Property. Some common values are as follows:

- **DdsKey** - used on the function key buttons.
- **MessageLine** - used on the message line used for displaying input errors.
- **DdsErrorReset** - used on the button shown when there are input errors.
- **DdsRecord** - used as a Div wrapping record.
- **DdsSflMsgField** - a character field used for Message Subfiles.
- **DdsCharField** - used for all other character fields.
- **DdsDecField** - used for all decimal fields.
- **DdsConstant** - used for literal fields.
- **DdsDateField** - used for date fields.

If a field is found to be in error, (badly typed input or user turned indicator on marking it as in error), the value of the CssClass property is suffixed with **_Error**.

Since users can modify the CssClass property, assume a character field has been modified to have a class name of MyClass, then when found in error, it would be

generated to the browser with a style of **class=MyClass_Error**. Refer to **Display File Template and Cascading Style Sheet Considerations** for more information on the Common.css file containing the named styles used to set the CssClass Property.

In addition, a right justified field will have **_Right** appended, while those that are output only have **_OutputOnly** appended. Always refer to the latest Cocoon.css file for the current styles.

Appendix I shows individual display file keywords and their level of support.

Printer Files

Two techniques are employed by RPG report applications to describe the layout of the report; **externally-described printer files** and the use of **O-Specs** to internally describe the layout. Monarch provides two migration agents to deal with these two techniques. Both agents target the DataGate Printer File facilities. These facilities consist of:

- Print Files
- Print File Designer
- Data-aware controls
- Render engine

The **printer O-Spec agent** takes the RPG O-Specs as input and generates a DataGate Print File. Print File Overflow Areas are also supported.

The **print file agent** takes DDS specifications as input and creates a DataGate Print File. **Appendix II** specifies the level of support provided by Monarch for Printer file DDS keywords.

Database Files

If data files are to be moved to SQL Server to be used at run time via DataGate for SQL Server (DSS), then DSS restrictions apply to the migrated application.

Even though DSS tries to make *SQL Server* look like DB2/400, there are several features that can not be implemented in a totally transparent fashion.

The next sections deal with all the issues that will need your attention, but the following items are probably the ones with the most impact for many of you.

Dealing with application using multi-member files

DataGate for SQL Server (DSS) does not implement the concept of a multi-member file. When migrating an application that makes use of a multi-member file, it is necessary to modify the application to use multiple files instead of multiple members. Instead of using the add member methods, one must use the copy file methods.

Under DSS, each file is implemented by a table or view depending upon whether the file is physical or logical. On the iSeries, a physical file has several attributes such as the record format definition, which is an order collection of fields, an optional key set at zero or more member that are the actual containers of data records. By contrast, a SQL Server table also defines a collection of columns and contains a set of rows, but the concept of data members is nonexistent.

For illustration purposes, let us assume that the application uses a file called **Sales** with one member for each month of the year labeled January through December. Furthermore, at the beginning of each month, the application adds a new member to the file where the sales for the month are to be accumulated.

One approach is to create a file with no data that will serve as the model for all the files that will contain the actual data taking the place of the multiple members. Let's call the model file **Sales__Model**. For each member, a copy of the model file is created using the member name as a suffix to make the name unique.

The application uses the appropriate suffixed file to store data, relegating the model file only as a template for the record format and key description. In our example, at the start of a new month, say September, we would copy Sales__Model to Sales_September. Notice that by using a double underscore for the Model file and a single underscore to separate the file name from the suffix (member name) all related files list together under Database Managers.

If there are logical members associated with the application, a similar approach must be followed taking into account that the base file will have to be modified to accommodate the corresponding new combined file_member name. Also, notice that the longer concatenated name will most likely exceed the 10 character limit imposed by OS/400, but allowed under DataGate for SQL Server.

It is important to use DataGate facilities to create the suffixed files because a physical file is more than just a plain table and a logical file is more than just a view. Special consideration has to be given to the key that ends up being an index on the table, and there are extended properties that have to be added to the table/view to preserve other attributes like column headings and text.

> ***Note:*** *Print files, which are typically multi-format, are fully supported in DSS.*

Logical Field Restrictions

There are two restrictions on the usage of logical fields when they change the name or the type of their corresponding base physical field. When the field is retype, most typically because the field is a concatenation or substring of the physical, then the field becomes read-only. A logical field, whose name has changed from its physical base field, cannot be used as a key field in the logical file.

Unlocking Records

This is probably the most demanding area of application adaptation. The problem arises in two areas: using the Unlock keyword on the read operations and on the implementation of the operation Unlock.

DDS uses *SQL Server* Cursors to implement file access. When a file is opened for update, it is not possible to tell *SQL Server* to not lock the record on a read, so this option is not valid for files opened for Update. You have two options to solve this problem: Declare a second instance of the file marked as input only and use it wherever the NoLock option was given on a read/chain. The other alternative is to leave the NoLock and follow the **READ/CHAIN** with an unlock. However, this option has the problems stated in the next paragraph.

The other problem is in the use of the Unlock operation. Unlock leaves the cursor in a no-position state, meaning you can not perform a subsequent read (next/previous) without repositioning the file with a SET (SETLL/SETGT) or CHAIN.

Appendix I - DDS Display File Keywords

The following tables specify the level of support provided by Monarch for Display file DDS keywords.

Display Field-Level Keywords

Supported	Supported in future release	Not applicable	Unsupported
ALIAS	BLANKS	AUTO	CHRID
CHANGE	CHGINPDFT	BLKFOLD	DUP
CHECK	CHKMSGID	CMP	EDTMSK
COLOR	DLTCHK	CNTFLD	ENTFLDATR
COMP	DLTEDT	OVRATR	NOCCSID
DATE	DFTVAL	OVRDTA	
DATFMT	FLDCSRPRG	PUTRETAIN	
DATSEP	FLTFIXDEC	WRDWRAP	
DFT	FLTPCN		
DSPATR(1)	HTML		
EDTCDE	INDTXT		
EDTWRD	MAPVAL		
ERRMSG	MSGCON		
ERRMSGID	VALNUM		
LOWER			
MSGID			
RANGE			
REFFLD			
SYSNAME			
TEXT			
TIME			
TIMFMT			
TIMSEP			
USER			
VALUES			

(1) See table of display attributes supported later in this section.

Display Record-Level Keywords

Supported	Supported in future release	Not applicable	Unsupported
ALTNAME	CHGINPDFT	OVRATR	ALARM
CAnn	CLRL	OVRDTA	ALWGPH
CFnn	HELP	PUTOVR	ALWROL
CHANGE	INDTXT	PUTRETAIN	ASSUME
CHECK	LOGINP	RMVWDW	BLINK
CSRLOC	LOGOUT	WDWBORDER	CCSID
ERASE	PROTECT	WRDWRAP	CLEAR
OVERLAY	VALNUM		CSRINPUT
PAGEDOWN			CSRINPONLY
PAGEUP			DSPMOD
PRINT			ENTFLDATR
RTNCSRLOC (2)			ERASEINP
RTNDTA			FRCDTA
ROLLDOWN			GETRETAIN
ROLLUP			HOME
SETOF(F)			INVITE
TEXT			INZINP
VLDCMDKEY			INZRCD
WDWTITLE			KEEP
WINDOW			LOCK
			MDTOFF
			MSGALARM
			RETCMDKEY
			RETKEY
			RETLCKSTS
			SLNO
			UNLOCK
			USRDFN

(2) Support is not provided for cursor positioning **within** a field.

Display File-Level Keywords

Supported	Supported in future release	Not applicable	Unsupported
CAnn	ALTHELP	INDARA	ALTPAGEDWN
CFnn	CHGINPDFT	WRDWRAP	ALTPAGEUP
CHECK	DSPRL		ALWGPH
ERRSFL	HELP		CLEAR
PAGEDOWN	INDTXT		CSRINPONLY
PAGEUP	MSGLOC		DSPSIZ
PRINT	VALNUM		ENTFLDATR
REF			HOME
ROLLDOWN			INVITE
ROLLUP			MSGALARM
VLDCMDKEY			OPENPRT
WDWBORDER			PASSRCD
			PRINT(Lib)/File)
			USRDSPMGT

Subfile Keywords

Supported	Supported in future release	Unsupported
SFL	SFLDROP	SFLCSRPRG
SFLCLR	SFLFOLD	SFLDLT
SFLCTL	SFLINZ	SFLEND
SFLCSRRRN	SFLMODE	SFLENTER
SFLDSP	SFLRNA	SFLLIN
SFLDSPCTL		SFLROLVAL
SFLMSG		SFLSCROLL
SFLMSGID		
SFLMSGKEY		
SFLRCDNBR		
SFLSMGRCD		
SFLNXTCHG		
SFLPAG		
SFLPGMQ		
SFLSIZ		

Display Attributes for DSPATR Keyword

Supported	Supported in future release	Unsupported
ND-Non Display	HI-High Intensity	BL-Blinking Field
PC-Position Cursor		CS-Column Separator
PR-Protect		MDT-Set Change Data Tag
		OID-Operator Identification
		SP-Select by Light Pen
		RI-Reverse Image
		UL-Underline

Appendix II - DDS Print File Keywords

The following tables specify the level of support provided by Monarch for Print file DDS keywords.

Printer Field-Level Keywords

Supported	Supported in future release	Not applicable	Unsupported
ALIAS	FLTFIXDEC	BLKFOLD	CDEFNT
BARCODE	FLTPCN		CHRID
COLOR	INDTXT		CHRSIZ
DATE	TEXT		CPI
DATFMT			CVTDTA
DATSEP			DLTEDT
DFT			DTASTMCMD
EDTCDE			FNTCHRSET
EDTWRD			PRTQLTY
FONT (1)			TRNSPY
FONTNAME			TXTRTT
HIGHLIGHT			
MSGCON			
PAGNBR			
POSITION (2)			
REFFLD			
SKIPA			
SKIPB			
SPACEA			
SPACEB			
TIME			
TIMFMT			
TIMSEP			
UNDERLINE			

(1) Font "PointSize" height value supported only.

(2) Position supported when position-down and position-across are expressed as constants.

Printer Record-Level Keywords

Supported	Supported in future release	Not applicable	Unsupported
CPI	BOX	OVERLAY	CDEFNT
ENDPAGE	DRAWER		CHRSIZ
FONTNAME	DUPLEX		DFNCHR
HIGHLIGHT	INDTXT		DOCIDXTAG
LIPPI	LINE		DTASTMCMD
PAGSEG	OUTBIN		ENDPAGGRP
PRTQLTY	TEXT		FNTCHRSET
SKIPA			FONT
SKIPB			FORCE
SPACEA			GDR
SPACEB			INVDTAMAP
			INVMMAP
			PAGRTT
			STRPAGGRP
			ZFOLD

Printer File-Level Keywords

Supported	Supported in future release	Not applicable	Unsupported
REF	INDTXT	INDARA	DFNCHR
			FNTCHRSET
			SKIPA
			SKIPB

Filename: Consideration for Migration New.doc
Directory: \\Rebuild\deliver\help\ASNA Visual Studio 2005 Products\8.1\Monarch 3.1\Manuals
Template: C:\Documents and Settings\agardner\Application Data\Microsoft\Templates\Normal.dot
Title: Considerations for Migration for ASNA Monarch 3.1
Subject:
Author: Anne Gardner
Keywords:
Comments:
Creation Date: 8/9/2007 4:02:00 PM
Change Number: 3
Last Saved On: 9/7/2007 11:48:00 AM
Last Saved By: Anne Gardner
Total Editing Time: 3 Minutes
Last Printed On: 9/7/2007 11:49:00 AM
As of Last Complete Printing
Number of Pages: 16
Number of Words: 3,623 (approx.)
Number of Characters: 18,917 (approx.)