

z/OS Automatic Restart Manager

z/OS Automatic Restart Manager

Introduction

ARMWRAP Program



Frank Kyne
Susan Greenlee



International Technical Support Organization

z/OS Automatic Restart Manager

January 2002

Take Note! Before using this information and the product it supports, be sure to read the general information in "Special notices" on page v.

First Edition (January 2002)

This edition applies to ???Insert Version, Release Number??? of ???Insert Program Name, Program Number ???-???? for use with the ???Insert Operating System ???

This document created or updated on January 28, 2002.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2002. All rights reserved.

Note to U.S Government Users - Documentation related to restricted rights - Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

- Special notices** v

- IBM trademarks** vi

- Preface** vii
 - The team that wrote this Redpaper vii
 - Notice vii
 - Comments welcome. vii

- Summary of changes** ix
 - January 2002, First Edition ix

- Chapter 1. ARMWRAP - The ARM JCL Wrapper** 1
 - 1.1 Modifying the application JCL 2
 - 1.1.1 Example 1- Application with no dependencies 3
 - 1.1.2 Example 2 - Application with dependencies on other elements 4
 - 1.1.3 Example 3 - Application that is a prerequisite for other elements 4
 - 1.2 Setting up the security definitions 5
 - 1.3 ARMWRAP Messages and Codes 6

- Appendix A. Additional material** 7
 - Locating the Web material 7
 - Using the Web material 7
 - System requirements for downloading the Web material 7
 - How to use the Web material 8

- Related publications** 9
 - IBM Redbooks 9
 - Other resources 9
 - Referenced Web sites 9
 - How to get IBM Redbooks 9
 - IBM Redbooks collections. 9

- Index** 11

Special notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this document was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.


The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

IBM trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

e (logo)[®] 
GDPS[™]
IBM[®]
MVS/ESA[™]
MVS[™]

OS/390[®]
Parallel Sysplex[®]
RACF[®]
Redbooks (logo)[™] 
Redbooks[™]

S/390[®]
SP[™]
z/OS[™]

Other company trademarks

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

This Redpaper provides an introduction to the function of the MVS Automatic Restart Manager (ARM), for those not already familiar with this system component. It discusses how to set up an ARM policy, using an actual policy as an example.

It also provides information about how to use the ARMWRAP program to extend support for ARM to programs that have not been coded to include this support.

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Frank Kyne is a Senior I/T Specialist at the International Technical Support Organization, Poughkeepsie Center. He has been an author of a number of other Parallel Sysplex redbooks. Before joining the ITSO three years ago, Frank worked in IBM Global Services in Ireland as an MVS Systems Programmer.

Susan Greenlee worked in z/OS development intermittently for 20 years, with the most recent 4 years in Parallel Sysplex Development. Previous development areas include Transaction Trace, Component Trace, GTF, System Trace, and Reconfiguration. In her current position, Susan does technical support marketing for Linux on zSeries and manages the S/390 University program. .

Thanks to the following people for their contributions to this project:

Richard Conway
International Technical Support Organization, Poughkeepsie Center

Joe Griesemer
IBM Poughkeepsie

David Surman
IBM Poughkeepsie

Notice

This publication is intended to help Systems Programmers to implement and exploit the MVS Automatic Restart Manager and the supporting ARMWRAP program. The information in this publication is not intended as the specification of any programming interfaces that are provided by z/OS. See the PUBLICATIONS section of the IBM Programming Announcement for z/OS for more information about what publications are considered to be product documentation.

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this Redpaper or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:
ibm.com/redbooks
- ▶ Send your comments in an Internet note to:
redbook@us.ibm.com
- ▶ Mail your comments to the address on page ii.

Summary of changes

This section describes the technical changes made in this edition of the paper and in previous editions. This edition may also include minor corrections and editorial changes that are not identified.

Summary of Changes
for z/OS Automatic Restart Manager
as created or updated on January 28, 2002.

January 2002, First Edition

This revision reflects the addition, deletion, or modification of new and changed information described below.

New information

- ▶
- ▶

Changed information

- ▶
- ▶



ARMWRAP - The ARM JCL Wrapper

As originally delivered with MVS/ESA V5.2, any program that wished to exploit ARM had to be modified to use the ARM API (IXCARM). They also had to be in system mode (system key or in supervisor state) when doing so. As a result, programs that wished to use ARM required code changes. However, it could be difficult to get these changes made to every application that you wish to use ARM. In addition, many of the tasks that you would like to use ARM are bought-in products to which you do not have the source. Also, installations generally prefer that their application programs do not run authorized, but this was a requirement in order to exploit ARM.

To resolve this problem, IBM changed ARM so that most ARM calls can be made by unauthorized programs. Also, a program called ARMWRAP was developed that provides the ability to exploit ARM without having to make changes to the application code. The support that allows ARM APIs to be invoked by unauthorized users under SAF control was added by APAR OW32480. This support is included in the base at OS/390 V2R9 and above.

There are five functions that can be specified on the IXCARM macro, namely:

REGISTER	ARM is made aware of the presence of the calling program, and will automatically restart that program should there be a failure.
WAITPRED	ARM will delay the initialization of this program until all predecessor programs have completed initialization.
READY	ARM is to mark this program as READY. Any other ARM-controlled programs that are waiting to use the service of this program can now be allow to initialize.
DEREGISTER	This program has completed successfully and should no longer be controlled by ARM
ASSOCIATE	Requests that this program be associated with another ARM-registered element for takeover or restart processing.

Of these five functions, all except the ASSOCIATE function can be used via ARMWRAP.

The ARMWRAP program issues the ARM APIs on behalf of the application. You control when the application invokes the ARM APIs based on where you place the JCL steps that execute ARMWRAP. ARMWRAP executes unauthorized, that is, it is a Key 8 problem program. In order to have your application exploit ARM via the ARMWRAP program, you need to set up the security environment for ARMWRAP and change the JCL for the job or started task to invoke the ARMWRAP program.

If you prefer that the application is not marked as READY to ARM until a particular message is issued, you can use the ARMREADY MPF exit that is supplied as part of the ARMWRAP package. The ARMREADY MPF exit will issue the IXCARM macro with the REQUEST=READY parameter. By issuing the READY request after the application issues an “initialization complete” type of message, the restart of other elements (that is, other applications that exploit ARM) in the same restart group will be appropriately synchronized with your application. Example 1-2 on page 4 and Example 1-3 on page 5 provide scenarios of the use of the MPF exit. You can use the ARMREADY as provided (that is, no customization is required) unless you need to interrogate the contents of the message.

Using the ARMWRAP program rather than changing the code to issue the ARM APIs may be sufficient for your application. However, there may still be situations where you need to modify an application to fully support ARM. The ability of the application to multi-thread and the performance benefits of issuing the IXCARM WAITPRED and IXCARM READY calls precisely when needed may be a compelling or required reason to change source code. Also, if you need to exploit IXCARM macro parameters that cannot be specified as parameters to the ARMWRAP program, you will need to change the application program source code. Even if you do decide to modify the application to call ARM directly, the ARMWRAP program can still be used to test an application’s exploitation of ARM before making those changes.

As for all ARM exploitations, ARM couple data sets are required and an ARM policy must be activated. See *MVS Setting Up A Sysplex*, SA22-7625, for more information.

1.1 Modifying the application JCL

At a minimum, you need to add the JCL to execute ARMWRAP to register the application with ARM before the step that executes the application, and add JCL to execute ARMWRAP to deregister the application with ARM after the step that executes the application.

The format of the ARMWRAP JCL statement is:

```
//nnnnnnnn EXEC PGM=ARMWRAP,PARM=(
      REQUEST = REGISTER
            ,TERMTYPE = ALLTERM | ELEMTERM
                    ,ELEMENT = name (up to 16 characters)
                    ,ELEMENTTYPE = DEFAULT | name (up to 8 characters)
            ,READYBYMSG = Y|N
      REQUEST = WAITPRED
      REQUEST = DEREGISTER )
```

When using ARMWRAP, the first call for a given program *must* be a REQUEST=REGISTER. After all, if ARM is not aware of an element, it cannot do any management on behalf of that element. If you do not specify READYBYMSG=N on the REQUEST=REGISTER, ARMWRAP will only issue an IXCARM REQUEST=REGISTER—the default is READYBYMSG=Y, meaning that ARMWRAP assumes that programs will issue an “initialization complete” type of message. If you do specify READYBYMSG=N, ARMWRAP will issue the IXCARM REQUEST=REGISTER call, followed immediately by an IXCARM REQUEST=READY. If you specify READYBYMSG=Y (or let it default), the IXCARM REQUEST=READY will be issued

by the ARMREADY MPF exit later on, when the application issues the “initialization complete” message. REQUEST, TERMTYPE, ELEMENT, and ELEMTYPE are IXCARM macro parameters. See *MVS Programming: Sysplex Services Reference*, SA22-7618, for more information on these parameters.

The following IXCARM parameters are not supported by ARMWRAP:

- ▶ EVENTEXIT
 - Use an ARM Element Restart Installation Exit if needed.
- ▶ ANSAREA
- ▶ REQUEST=ASSOCIATE
- ▶ STARTTXT
 - Use the RESTART_METHOD statement in the ARM policy if needed.
- ▶ RESTARTTIMEOUT
 - Use the RESTART_TIMEOUT statement in the ARM policy if needed.
- ▶ ELEMBIND

1.1.1 Example 1- Application with no dependencies

Example 1 works well for an application that is in a restart group without other elements or is in a restart group with other elements where there are no dependencies by those elements on the application that you want to restart, that is, your application has a level number in the restart group that is equal to or higher than the highest level number of all other elements in the restart group. Note that higher level elements are dependent on lower level elements in the same restart group and there can be more than one element with the same level number in a restart group.

In this example, once ARMWRAP runs, MYAPPL will be restarted if it fails, or if the system it is running on fails, assuming the active ARM policy does not prevent the restart. ARMWRAP will issue the IXCARM macro with the REGISTER parameter to register the element MYAPPLELEMENT with ARM immediately followed by the IXCARM macro with the READY parameter to tell ARM that MYAPPL is ready to provide services to other applications in the restart group. After MYAPPL completes execution, ARMWRAP is executed again to issue the IXCARM macro with the DEREGISTER parameter to notify the system that restarts are no longer required for the application. If you forget this step, ARM will restart MYPROC as soon as MYAPPL terminates.

Example 1-1 Sample ARMWRAP JCL for application with no dependencies

```
//MYPROC PROC ...
/* Register element 'MYAPPLELEMENT' element type 'APPLTYPE' with ARM
/* Requires access to SAF FACILITY IXCARM.APPLTYPE.MYAPPLELEMENT
//ARMREG EXEC PGM=ARMWRAP,
//      PARM=('REQUEST=REGISTER,READYBYMSG=N,',
//          'TERMTYPE=ALLTERM,ELEMENT=MYAPPLELEMENT,',
//          'ELEMTYPE=APPLTYPE')
//MYAPPL EXEC PGM= MYAPPL,PARM=(&MYPARM1,&MYPARM2)
/* For normal termination, deregister from ARM
// ARMDREG EXEC PGM=ARMWRAP,
//      PARM=('REQUEST=DEREGISTER')
//SYSABEND DD SYSOUT=*
```

1.1.2 Example 2 - Application with dependencies on other elements

When a system failure occurs, ARM restarts all the elements in the restart group in parallel and synchronizes initialization and service dependencies for you. If the application MYAPPL is an element of a restart group and the services of one or more elements in the restart group must be available for MYAPPL to *initialize*, then use the ARMWRAP program with the WAITPRED parameter to coordinate element restarts within the group. When ARMWRAP specifies the WAITPRED parameter, ARM synchronizes initialization so that other elements in the restart group that have a lower level number are in the ARM ready state before MYAPPL executes. This step is not required if MYAPPL uses the services of one or more elements in the restart group during its normal processing, that is, after it has initialized, because ARM coordinates all elements within the restart group through the READY parameter of the IXCARM macro.

Because the IXCARM REQUEST=WAITPRED cannot be issued after the IXCARM REQUEST=READY call, in this scenario you must use READYBYMSG=Y on the REQUEST=REGISTER call to stop that step from issuing the READY call. Because the REQUEST=REGISTER will not issue the IXCARM REQUEST=READY, you must set up an MPF exit to issue the IXCARM macro with the REQUEST=READY parameter and MYAPPL must issue an initialization or start up message for the MPF exit to get control. “Example 3 - Application that is a prerequisite for other elements” on page 4 contains more information about the MPF exit.

Example 1-2 Sample ARMWRAP JCL for application with dependencies

```
//MYPROC PROC ...
/* Register element 'MYAPPLELEMENT' element type 'APPLTYPE' with ARM
/* Requires access to SAF FACILITY IXCARM.APPLTYPE.MYAPPLELEMENT
//ARMREG EXEC PGM=ARMWRAP,
//      PARM=('REQUEST=REGISTER,READYBYMSG=Y',
//           'TERMTYPE=ALLTERM,ELEMENT=MYAPPLELEMENT,',
//           'ELEMTYPE=APPLTYPE')
/*
/* On a restart, wait for any predecessors in the ARM restart group
//ARMPRED EXEC PGM=ARMWRAP,
//      PARM=('REQUEST=WAITPRED')
/*
/* When MYAPPL issues msg MYAPP101, the MPF exit issues IXCARM READY
//MYAPPL EXEC PGM= MYAPPL,PARM=(&MYPARM1,&MYPARM2)
/*
/* For normal termination, deregister from ARM
//ARMDREG EXEC PGM=ARMWRAP,
//      PARM=('REQUEST=DEREGISTER')
//SYSABEND DD SYSOUT=*
```

1.1.3 Example 3 - Application that is a prerequisite for other elements

When a system failure occurs, ARM restarts all the elements in the restart group in parallel and synchronizes initialization and service dependencies for you. If the application MYAPPL is an element of a restart group and the services of MYAPPL must be available for *other* elements in the restart group to *initialize*, then MYAPPL needs to transition to the ARM READY state *after* it completes initialization. The trigger to get to the ARM READY state is via the ARMREADY MPF exit that gets control when MYAPPL issues an “initialization complete” type of message. For example, if MYAPPL initializes and then issues message ‘MYAPP101 INITIALIZATION COMPLETE’, the ARMREADY MPF exit that you set up to get control when the message is issued will issue the IXCARM macro with the READY parameter. The MPF exit executes authorized in the message issuer’s address space under the unit of work that issued the message. The IXCARM macro with the READY parameter must be issued from

the registering address space. Because the IXCARM REQUEST=READY call does not contain the ELEMENT or ELEMENTYPE, you only need to modify the MPF exit if you need to interrogate the message text for the message id, otherwise, you can link and use it as is. The additional material associated with this document contains an unloaded PDS that contains an SMP Usermod for the ARMREADY exit, as well as the Assembler source for the exit should you wish to modify it.

ARM will synchronize initialization so that other elements in the restart group that have a higher level number than MYAPPL are moved to the ARM-ready state only after the ARMREADY MPF exit for MYAPPL executes. This step is not required if MYAPPL services are used by one or more elements in the restart group during its normal processing. The coordination of elements within the group during normal processing is controlled by the READY parameter of the IXCARM macro.

Example 1-3 Sample ARMWRAP JCL for application that is a pre-req for other elements

```
//MYPROC PROC ...
/* Register element 'MYAPPLELEMENT' element type 'APPLTYPE' with ARM
/* Requires access to FACILITY IXCARM.APPLTYPE.MYAPPLELEMENT
//ARMREG EXEC PGM=ARMWRAP,
//      PARM=('REQUEST=REGISTER,READYBYMSG=Y',
//            'TERMTYPE=ALLTERM,ELEMENT=MYAPPLELEMENT,',
//            'ELEMENTYPE=APPLTYPE')
/*
/* When MYAPPL issues msg MYAPP101, the MPF exit issues IXCARM READY
//MYAPPL EXEC PGM= MYAPPL,PARM=(&MYPARM1,&MYPARM2)
/*
/* For normal termination, deregister from ARM
// ARMDREG EXEC PGM=ARMWRAP,
//      PARM=('REQUEST=DEREGISTER')
//SYSABEND DD SYSOUT=*

MPFLSTxx entry for message MYAPP101.....
MYAPP101,SUP(NO),USEREXIT(ARMREADY)
```

1.2 Setting up the security definitions

Since ARMWRAP runs unauthorized, you need to do the following to set up the necessary security:

- ▶ Define the resource 'IXCARM.element_type.element' in the FACILITY class
 - In the MYPROC example we have been using, the resource to be defined is IXCARM.APPLTYPE.MYAPPLELEMENT.
 - If the ELEMENTTYPE parameter is not specified on the ARMWRAP execute statement, then use the string 'DEFAULT'.
- ▶ Authorize jobs and started tasks with UPDATE access to this resource.
 - In the example we have been using, MYPROC needs to have UPDATE access to the resource.

Example 1-4 on page 5 contains a REXX exec that will issue all the RACF commands to define the required profile and accesses. Note that this example assumes that the started task name is MYPROC and that it has not been previously defined to RACF. The example defines a new userid called WRAPPER and associates that with the MYPROC started task.

Example 1-4 Sample REXX exec to set up security definitions

```

REXX Example
/* Activate the FACILITY and STARTED classes */
'SETR CLASSACT(FACILITY STARTED) '

/* Add user "WRAPPER" with a logon proc SPF130, ACCTNUM, and SIZE */
'AU (WRAPPER) TSO(PROC(SPF130) ACCTNUM(nnnnnnnn) SIZE(20000)) '

/* Define PROC "MYPROC" to user "WRAPPER" */
'RDEFINE STARTED MYPROC.** STDATA(USER(WRAPPER) GROUP(SYS1) TRACE(YES))

/* Define resource IXCARM.APPLTYPE.MYAPPLELEMENT with a universal access of NONE */
'RDEF FACILITY IXCARM.APPLTYPE.MYAPPLELEMENT UACC(NONE) '

/* Permit resource "IXCARM.APPLTYPE.MYAPPLELEMENT" to user "WRAPPER" with update access */
'PERMIT IXCARM.APPLTYPE.MYAPPLELEMENT CLASS(FACILITY) ID(WRAPPER) ACC(UPDATE) '

/* refresh in-storage classes */
'SETR RACLIST(FACILITY STARTED) REFRESH'

```

1.3 ARMWRAP Messages and Codes

The following syntax error messages may be issued by ARMWRAP:

```

ARMWRAP WRAPPER SYNTACTICAL ERROR -
    Duplicate REQUEST specified
    Duplicate TERMTYPE specified
    Duplicate ELEMENT specified
    Duplicate ELEMTYPE specified
    Duplicate READYBYMSG specified
    TERMTYPE is either ELEMTERM or ALLTERM
    REQUEST is either WAITPRED or (DE)REGISTER
    ELEMENT must be <= 16 characters
    ELEMTYPE must be <= 8 characters

```

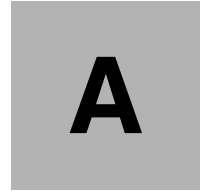
The following functional error messages may be issued by ARMWRAP:

```

ARMWRAP IXCARM RC = xxxx RSN = xxxx

    RC = xxxx, refer to the IXCARM macro
    RSN = xxxx, refer to the IXCARM macro

```



Additional material

This Redpaper refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this Redpaper is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

<ftp://www.redbooks.ibm.com/redbooks/REDP0173>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the Redpaper form number, REDP0173.

Using the Web material

The additional Web material that accompanies this Redpaper includes the following files:

<i>File name</i>	<i>Description</i>
ARMWRAP.zip	PDS in IEBCOPY UNLOAD format containing usermods to install the ARMWRAP and ARMREADY programs, and the Assembler source for the ARMREADY MPF exit.

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: 1MB minimum
Operating System: Any

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder. Then send the resulting file, in Binary format, to an LRECL 80, FB data set on z/OS. A TSO RECEIVE INDSN(hlq.filename) should then be issued to create the PDS.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 9.

- ▶ *OS/390 Parallel Sysplex Configuration Volume 1: Overview*, SG24-5637

Other resources

These publications are also relevant as further information sources:

- ▶ *z/OS MVS Programming: Sysplex Services Guide*, SA22-7617
- ▶ *z/OS Setting up a Sysplex*, SA22-7625

Referenced Web sites

These Web sites are also relevant as further information sources:

- ▶ Description1
<http://www.redbooks.ibm.com/>

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

A

- ARMREADY MPF exit 2
- ARMWRAP program
 - example with dependencies 4
 - example with dependency 4
 - example with no dependencies 3
 - how to invoke 2
 - parameters 2
 - RACF definitions 5
 - support prerequisites 1
 - supported IXCARM functions 1
 - unsupported IXCARM functions 3

I

- IXCARM macro 1
 - ASSOCIATE function 1
 - calling from MPF exit 4
 - DEREGISTER function 1
 - READY function 1
 - REGISTER function 1
 - WAITPRED function 1

J

- JCL to execute ARMWRAP 2

M

- messages and codes 6

R

- RACF definitions for ARMWRAP 5
- Redbooks Web site 9
 - Contact us viii



z/OS Automatic Restart Manager



z/OS Automatic Restart Manager

Introduction

ARMWRAP Program

This Redpaper provides an introduction to the benefits of the z/OS Automatic Restart Manager and discusses how to set up an ARM policy to match your requirements.

The Redpaper also describes the ARMWRAP program that allows ARM support to be added to tasks that have not implemented ARM.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks