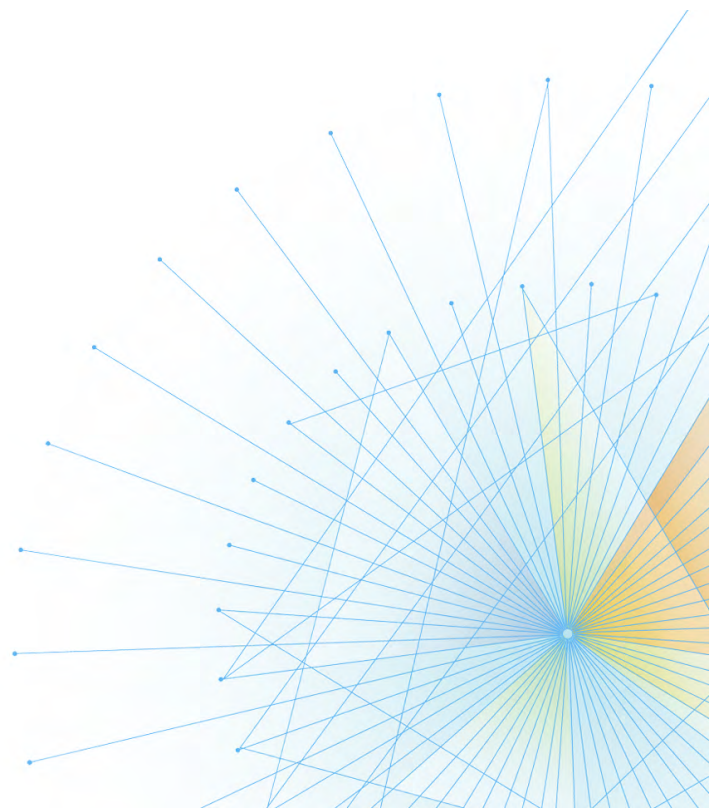




The Mainframe Software Partner For The Next 50 Years

ISPW Deploy Reference

Release 18.02



Please direct questions about ISPW
or comments on this document to:

Compuware Support Center

<https://go.compuware.com/>

This document and the product referenced in it are subject to the following legends:

Copyright 1984 - 2019 Compuware Corporation. All rights reserved. Unpublished rights reserved under the Copyright Laws of the United States.

U.S. GOVERNMENT RIGHTS-Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in Compuware Corporation license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable. Compuware Corporation.

This product contains confidential information and trade secrets of Compuware Corporation. Use, disclosure, or reproduction is prohibited without the prior express written permission of Compuware Corporation. Access is limited to authorized users. Use of this product is subject to the terms and conditions of the user's License Agreement with Compuware Corporation.

ISPW and ISPW Deploy are trademarks or registered trademarks of Compuware Corporation.

CICS, CICS TS, DB2, IBM, IMS, MVS, MVS/ESA, OS/390, RACF, VTAM, and z/OS are trademarks or registered trademarks of International Business Machines Corporation.

Adobe® Reader® is a trademark of Adobe Systems Incorporated in the United States and/or other countries.

All other company and product names are trademarks or registered trademarks of their respective owners.

Contents

Introduction	7
Related Publications	7
Online Documentation	8
Customer Solutions	8
ISPW Deploy Overview	9
ISPW Deploy Architecture	10
ISPW Deploy Concepts	11
Deployment Environment	11
Useful for Testing Environments	12
Multiple Implementation	12
Product Implementation	12
Deployment Environment Can Span Platforms	12
Deployment Sub-Environment	12
Deploy Sub-Environment	13
Further Examples	13
Deployment Targets and Systems	13
Deployment System	13
Deployment Target	13
Deployment Requests	14
Deployment Package	14
Deployment Types	14
How They Relate to ISPW Component Types	15
Example	15
Associating a Category	15
Deployment Categories	15
Why are they required?	15
How are they used?	15
Example	15
Initial Implementation	16
Deployment Domains	16
Why They are Needed	16
Deploy Maintenance Functions	19
Maintenance Menu	19
More Information	20
DS - Deployment Systems	20
Deployment System List Screen	20
Detail Screen	21
DD - Deployment Domains	22
Deployment Domain List Screen	22

Detail Screen	23
DC – Deployment Categories	23
Deployment Category List Screen	24
Detail Screen	24
DT – Deployment Types	25
Deployment Type List Screen	25
Detail Screen	26
DV – Deployment Environment	27
Environment Definition	27
Sub-Environment Definition	28
Defining Deployment to ISPW	35
Start with a Simple Deployment	35
Assumptions	35
1. Define a Deployment Category	35
2. Define Deploy Types	36
3. Associate Deploy Types to ISPW Types	36
4. Define Deploy System	37
5. Define a CT Server for Remote Deployment	37
6. Define a Warehouse for Staging	37
7. Define Deployment Domain	37
8. Assign Domain to Application	38
9. Define Deploy Environment	38
10. Add Targets to Deploy Sub-Environment	40
11. Add Deploy Type “Names”	42
12. Activate Deployment Environment	43
13. Indicate Deployment on Application	44
14. Indicate Implementation for Each Type	45
15. Specify Deploy Environment	45
16. Refresh Server	46
Deployment Setup Checklist	47
Using Deploy Categories	47
Effect on Deploy Packages	47
Specifying Different Categories	47
Making Use of Different Categories	48
Using Deploy Domains	48
Exploiting Domains	48
Define Multiple Domains	48
Assign Different Domains to ISPW Applications	48
Example	48
Adaptation – MVS Only	48
When would it be used?	48
How does adaptation satisfy this requirement?	49
Control Card Syntax	49
Examples	49
Ensure Adaptation is Performed	50

Advanced Implementation – All Platforms	50
Implementation Processes	51
Activation – All Platforms	51
Example of Use	51
Activation Processes	51
Deployment Processes	53
Overview	53
ISPW Standard Implementation	53
More Complex Implementation Requirements	53
Deployment Processes	53
Starting a User Process	53
ISPW Started Process	53
Deploy Process for MVS	54
Process Start	54
MVS API Call – PACKAGE START	55
MVS API Call – ITEM GET	58
MVS API Call – ITEM END	59
MVS API Call – PACKAGE END	60
MVS Deploy Log	61
Log Dataset	61
Logging to the Dataset	62
Keeping the Logs	62
Viewing the Logs	63
Deployment Process for Distributed Platforms	63
Distributed Deploy Definition Concepts	63
Deploy Activation Monitor	68
Monitoring	69
WZUDPAP – Writing to the Log	70
Example Usage for WZUDPAP	70
Example Log	70
Security	73
Overview	73
Security Enhancements	73
Deploy Reference Data	73
What has been added?	73
Component Transport Security	74
MVS STC	74
CT Messages	74
CT Identification	74
CT Security Token	74
User Functions	74
Protecting Deploy Requests	74
User Reference	77
Overview	77

Deployment Requests	77
Breaking a Request into Packages.	77
Deployment Phases	77
Deploy Request Queue.	78
Deploy Request Messages	81
Deploy Request Modify	82
Deploy Request Packages - Composite View.	83
Deploy Request Packages - Summary view	85
Deploy Request Packages - System View	86
Deploy Request Package – List Items	87
Deploy Request Package – View Details	90
Deploy Request Package – Fail Physical Package.	91
Task List Options and Display	93
Task List Status Area.	93
Task OP – DP	94
Task OP – DT	95
Operation History – Deploy Operations	97
Deploy Set Processing	98
Deploy Logging	99
Introduction	99
Intended Audience.	99
ISPW Services.	99
Background	99
Overview.	99
Features	99
Configuration	100
M.ER	100
New TSI Calls.	100
Processing Details	102
DB2 Table	102
Add Processing	103
Subsequent Deploys	103
Action D Processing.	104
Clean Up Processing	104

Introduction

This document introduces ISPW users to the new ISPW Deploy product. It gives details of the concepts and facilities from both an end-user and technical perspective. An outline of the contents of this document is given below by a short description of each chapter:

[“ISPW Deploy Overview”](#) — This chapter gives an overview of the ISPW Deploy product.

[“ISPW Deploy Concepts”](#) — This chapter describes the key concepts used by ISPW Deploy.

[“Deploy Maintenance Functions”](#) — This chapter contains details of the definition maintenance dialogs that are used to configure the ISPW Deploy product.

[“Defining Deployment to ISPW”](#) — This chapter contains details of the definition maintenance dialogs that are used to configure the ISPW Deploy product.

[“Deployment Processes”](#) — This chapter describes how to use the Deploy API to write external process for Implementation and Activation.

[“Security”](#) — This chapter describes the security aspects for ISPW Deploy.

[“User Reference”](#) — This chapter describes the User functions that are used to monitor and control Deployments.

[“Deploy Logging”](#) — This chapter describes the Deploy Logging feature.

Related Publications

- *ISPW Release Notes*: Overview of release features and any new ISPW information.
- *ISPW Installation and Configuration Guide*: Gives step-by-step instructions for the system programmer to configure, customize, and maintain ISPW. Refer to it when installing ISPW according to the *Compuware Installer Mainframe Products SMP/E Installation Guide*.
- *ISPW Interfaces Guide*: Describes ISPW’s external call interface, stand alone load modules, and use with DB2 programs, plans, packages, and generated DECLARE statements. Interfaces with Telon, QMF, and Natural are also documented.
- *ISPW Messages and Codes*: Documents the messages and codes for ISPW, including started task errors, abend codes, return codes, allocation errors, and ISPW CM errors.
- *ISPW Planning Guide*: Provides information for use in the early stages of an ISPW implementation. It contains only what is necessary for planning and initial installation.
- *ISPW Remote Server Guide*: Describes ISPW remote servers, Controlled Tasks run on platforms separate from the main ISPW Administration Server.
- *ISPW Technical Reference*: Provides detailed information on ISPW’s structure, terms and concepts, maintenance functions, processing, security, and other technical content.
- *ISPW User Guide*: Provides an overview for Applications and other Information Systems staff to use ISPW effectively.
- *ISPW Upgrade Guide, Release 4.4 to 17.02*: Describes the major differences between ISPW 4.4 and ISPW 17.02 and serves as a guide for the upgrade process between these versions.

- *ISPW Web Interface Installation and Configuration Guide*: Provides instructions on how to install the ISPW Web Interface. ISPW Web is an Internet-based application designed to be used on workstations or smartphones. The ISPW Web Interface uses a Web browser that enables you to remotely approve or reject ISPW actions as well as deploy software to both mainframe and distributed environments.

Online Documentation

The ISPW product installation package does not include the product documentation. Access the ISPW documentation from the Compuware Support Center website at <https://go.compuware.com> in the following electronic formats:

- Release Notes in HTML format
- Product manuals in PDF format
- Product manuals in HTML format.

The product documentation is available for viewing or downloading:

- View PDF files with the free Adobe Reader, available at <http://www.adobe.com>.
- View HTML files with any standard web browser.

Customer Solutions

Visit the Compuware Support Center, <https://go.compuware.com>, to find product documentation, knowledge articles, and other technical resources. You can open a case with the Customer Solutions team, order products, and much more.

Contact Customer Solutions by phone:

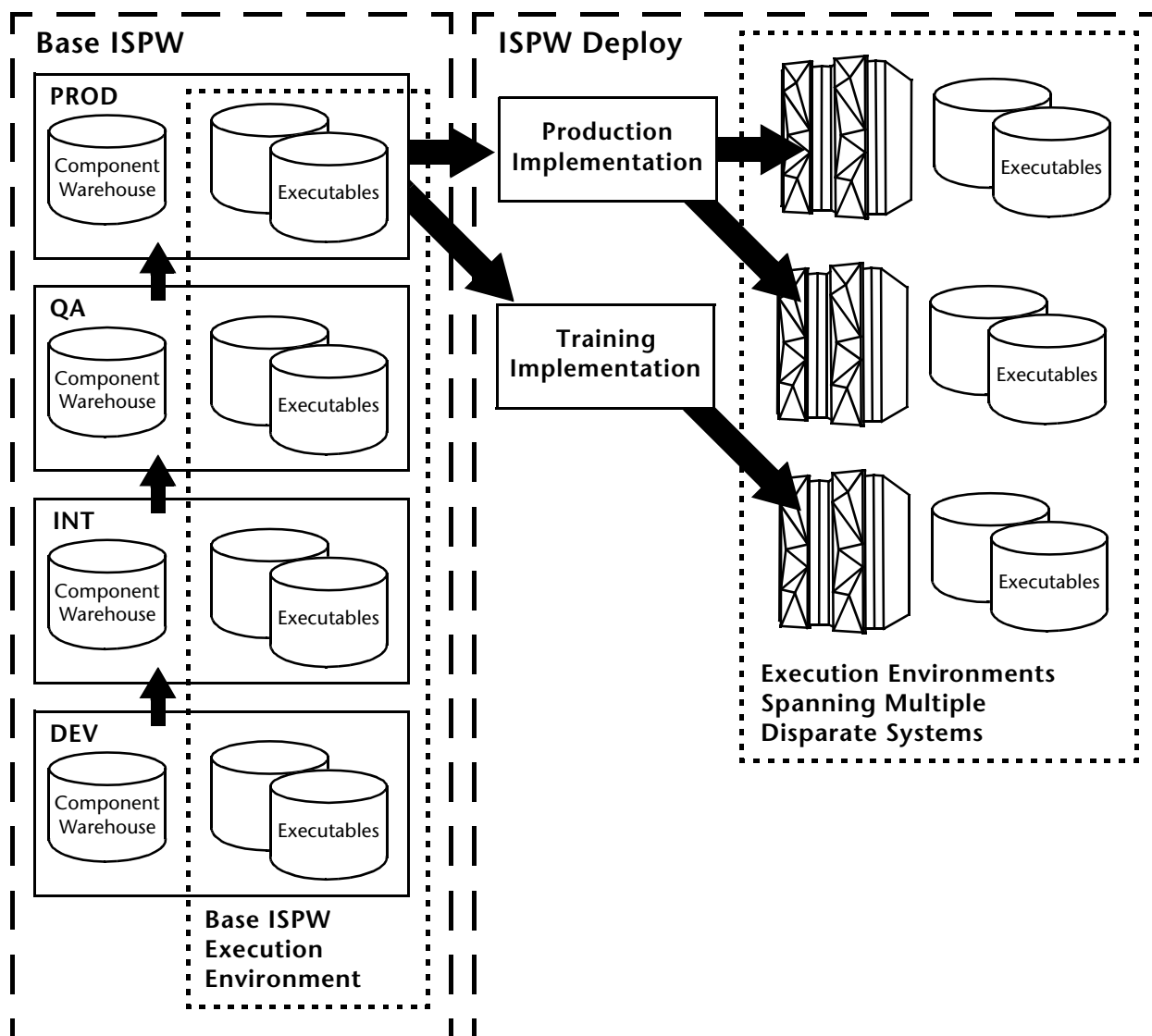
- USA and Canada: 1-800-538-7822 or 1-313-227-5444.
- All other countries: Contact your local Compuware office. Contact information is available at <https://go.compuware.com>.

Visit Compuware on the web at <http://www.compuware.com> for additional product information.

ISPW Deploy Overview

ISPW Deploy provides the capability to deploy software components into execution environments and is a companion product to base ISPW. [Figure 1](#) shows the interaction between ISPW Deploy and base ISPW.

Figure 1 ISPW Diagram



Base ISPW provides application life cycle management functions, which include software configuration, change and process management using a promotional model. The change life cycle is defined by a hierarchy of levels where each level represents a stage in the cycle.

Each ISPW level can be used as an execution environment that is directly managed as part of the change cycle so eliminating the need for additional steps to deploy into a separately managed execution environment.

This method of working has proved very successful as it has a number of advantages but it does not provide the complete solution in the following situations:

- Applications that have multiple releases that are operational in production.
- Applications that require multiple physical implementations in production, or any other level.
- Applications whose execution environments require the software to be distributed to other systems.
- Organizations who completely separate the development and delivery of application releases from the actual deployment into the execution environments.

ISPW Deploy provides the complete solution for these situations. It allows the separation of deployment from development or to keep close integration between the two activities. It does this by supporting the following methods of operation:

- Closely integrate with base ISPW so that the deployment to one or more execution environments is still managed as part of the change cycle.
- Loosely integrate with base ISPW so that applications releases developed in ISPW can be deployed with complete independence from the software change cycle.
- Independently perform outside of base ISPW to deploy any software product by directly registering it to ISPW Deploy Release 18.02.



ISPW Deploy 18.02 will only support the closely integrated method.

ISPW Deploy Architecture

The execution environments into which deployment is required are first defined to ISPW Deploy using the Deploy Maintenance functions, see [“Deploy Maintenance Functions”](#).

ISPW Deploy uses Deployment Requests as the mechanism for requesting and processing a software deployment. A Deployment Request specifies the set of software components to be deployed, the target execution environment and timing and other processing constraints.

ISPW Deploy will then process the request, according to constraints specified, by distributing the components to the target deployment systems and activating them. As a minimum this involves placing the component in the correct storage location but could also include adaptation of the component where necessary as well as initiating other processes required to make the component active.

For deployment integrated with base ISPW additional definitions are necessary in base ISPW to connect the applications being managed to the desired execution environments in ISPW Deploy.

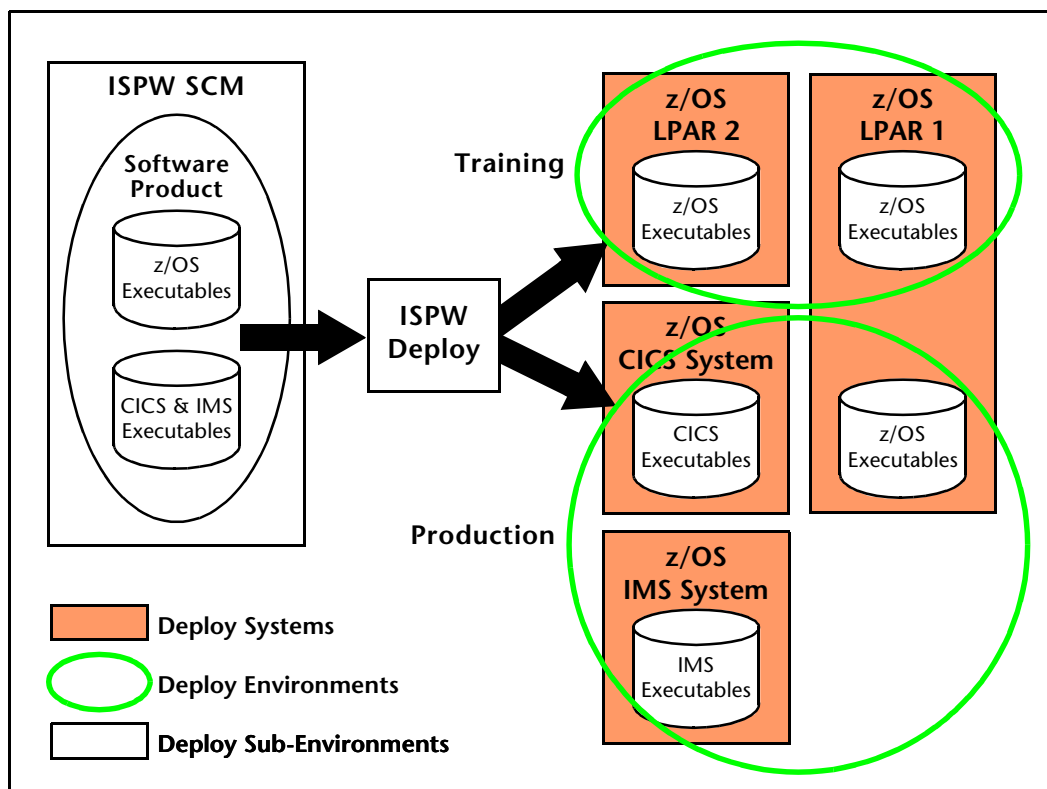
Deployment into an execution environment can be initiated automatically from base ISPW. When ISPW determines that deployment of a component using ISPW Deploy is necessary, it creates an appropriate deployment request that is then processed by ISPW Deploy. The status of the deployment request is reflected in the ISPW Task and Set status and the full details of the deployment request can be easily accessed from base ISPW.

ISPW Deploy Concepts

ISPW Deploy uses various concepts to manage the deployment of software. Some of the key concepts are explained in this chapter.

The ISPW Deploy diagram ([Figure 2](#)) depicts an example Deploy situation for the purpose of explaining the different concepts. Each section of this chapter makes reference to this diagram to help explain each concept.

Figure 2 ISPW Deploy Diagram



Deployment Environment

The Deploy Environment is the logical execution environment where a specific version of one or more software products is implemented.

In the ISPW Deploy diagram ([Figure 2](#)), the Deploy Environments are depicted in green.



Datasets that are part of a Deploy Environment must **not** be updated outside of ISPW Deploy. You should not edit them or run a program or utility that may update them. If you do so, you run the risk of overlaying the deployed dataset. Security rules should be in place to prevent users from updating Deployment Environment datasets directly.

Useful for Testing Environments

A software product may have several implementations to support running different versions such as, system test, acceptance test, production, etc. Each of these implementations would be a separate deployment environment.

Multiple Implementation

Several discrete implementations of the same version may also be required for different user groups. Each of these discrete implementations would also be a separate deployment environment.

Product Implementation

In many cases a deployment environment will just be a single implementation of a single product so it could be viewed merely as a product implementation. The deployment environment concept becomes essential where several products are implemented in the same execution environment that is managed operationally as a single environment.

Deployment Environment Can Span Platforms

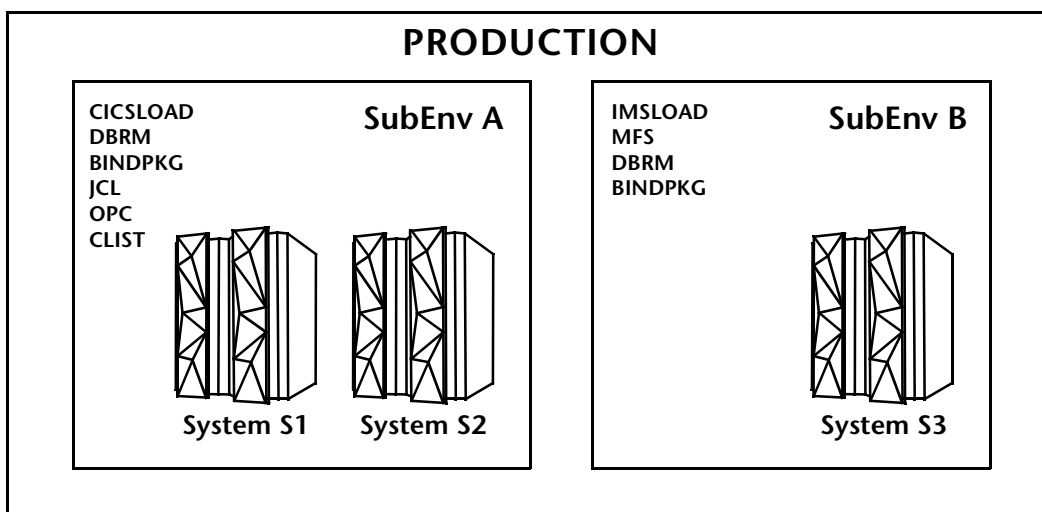
One important feature of the deployment environment is that it can represent a logical environment. A single implementation of a software product may require implementing components of the product in different execution environments on disparate platforms and systems. Where this all forms part of a single product implementation, this can be defined as just one deployment environment in ISPW Deploy.

Deployment Sub-Environment

As the deployment environment can represent a logical execution environment another concept is required to define the actual execution environments in which components of a product may need to be deployed. The deployment sub-environment is the concept used for this purpose.

The ISPW Deploy Environment diagram ([Figure 3](#) on page 12) depicts a Deploy Environment called "Production" that includes two sub-environments defined for the purpose of deploying the different component types to the correct targets.

Figure 3 Deploy Environment PRODUCTION



Deploy Sub-Environment

[Figure 3](#) shows that a Deploy Environment may be associated with multiple Sub-Environments. Each of these Sub-Environments is a combination of various Deployment Types and Systems. [Table 1](#) shows two different Sub-Environments that describe the deployment attributes for the different “components” that make up the Production System.

Table 1 Sub-Environments

Sub-Env	Description
A	Targets for the CICS and Batch related components
B	Targets for the IMS related targets

Note that this is simply an example, and different customers may use these relationships to describe their own particular deployment needs.

Further Examples

If a product implementation has some components that need to be deployed to an MVS CICS system and other components that need to be deployed to an IMS region then each of those environments would be defined as a deployment sub-environment.

Sub-environments may also be used to separate the major components of a product even when they are deployed on the same system. The only absolute requirement to use them is when components of a product are being deployed to disparate systems.

Deployment Targets and Systems

Some product implementations require deployment onto multiple system images to meet performance or service level agreements. In this case the deployment on each system image is exactly the same. The deployment target concept was designed to support this requirement.

In the ISPW Deploy diagram ([Figure 2](#) on page 11), the Deployment Systems are the gold boxes. An example of a Target would be the definition data required to implement sub-environment “Web” to system “Unix Webserver1”.

Deployment System

A deployment system identifies a particular system image on which many different products may be implemented. Usually this refers to some physical computer system, such as a Unix or Windows server, although for operating systems like MVS it may still be just a logical system image (LPAR).

Deployment Target

Defining deployment targets specifies the actual system images where the sub-environment is to be deployed. In other terms then, a Deployment Target is the instance of a deployment definition between a Sub-environment and System.

One or more deployment targets may be defined for a sub-environment and each one specifies a deployment system. The deployment sub-environment definition becomes a template (This is the level at which the target dataset names and process options are defined, which means that these names and options are the same for all systems defined as targets for that sub- environment).

Deployment Requests

A deployment request is an instruction to deploy one or more software component parts into a deployment environment. It contains a request header, which holds basic details about the request, and one or more deployment packages.

Deployment Package

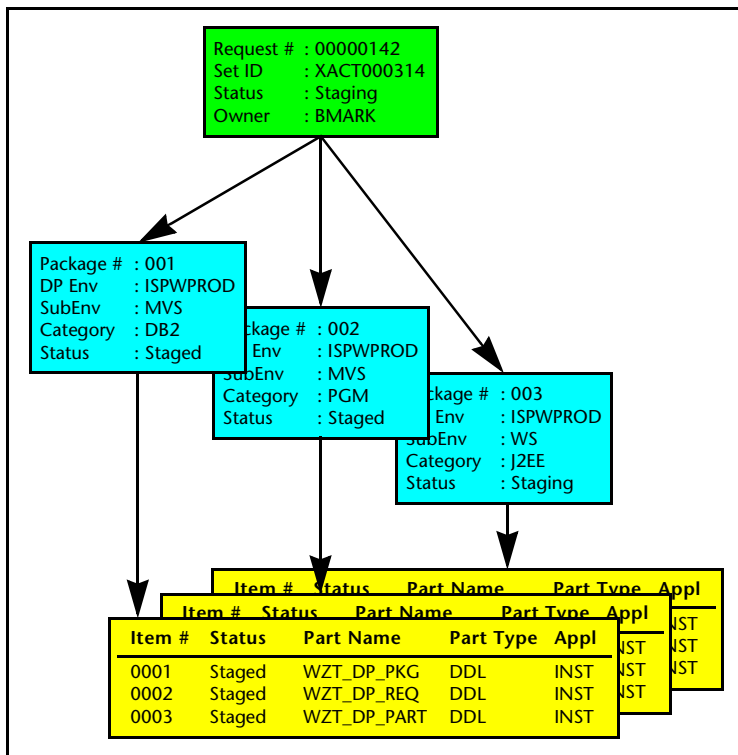
A deployment package is used to group all the software component parts that are destined for the same deployment sub-environment and have the same deployment category. A deployment package contains one or more deployment package items.

Deployment Package Item

Deployment package items specify software component parts that are to be deployed. They are deployed to all deployment targets that are defined for the sub-environment specified on the deployment package that contains the item.

[Figure 4](#) shows the make-up of a Deploy Request, with respect to Packages and Items.

Figure 4 Deploy Request Packages



Deployment Types

A Deployment Type identifies a component or file type for deployment. The Deployment Type is the level at which the target datasets are defined. The type is also used to determine the different activation processes required for a component.

How They Relate to ISPW Component Types

The Deployment Type corresponding to an ISPW Type/Class will be specified against “Stream Component Type”.

Example

For an ISPW type of LOAD that has a class of CICS, you could define a Deployment Type of LOADCICS (Deployment Types can have up to 8 characters in their names).

Associating a Category

For each Deployment Type, a Category can be specified. This allows for the grouping of Deployment Types for scheduling considerations. The Deployment Category is explained in the next section.

Deployment Categories

Deployment Categories are used to group Deployment Types together for activation or scheduling purposes.

Why are they required?

As deployment is concerned with the physical update of running (and even production) systems, there are constraints around the activation of different types of components. An example of this could be that CICS related components might require a window whereby the CICS region is down. Similarly, updates to a system requiring DB2 changes may need to be done in a specific window. Deployment Categories provide the mechanism to define these relationships between the Deployment Types.

How are they used?

Parts requiring deployment are packaged within a Deployment Request by the Deployment Category and Deployment Sub-Environment. These packages are the level at which scheduling and activation constraints can be calculated. You will be able to influence the timing of the activation based upon the different Categories.

Example

Take a simple example with the components listed in [Table 2](#),

Table 2 Deployment Categories

Type	Category
JCL	BATCH
LOADBAT	BATCH
LOADCICS	CICS

The BATCH category might have a constraint where each night there is a window where components can be activated. The CICS Category might have a window once a week.

If a Deployment Request contains components of all three types, then the calculation of the activation time will result in having to wait until the CICS window for all components such that the set of components as a whole get activated at the same time.

If a Deployment Request contains only JCL and LOADBAT type components, then they can be activated that night (if that is desired).

Initial Implementation

Initially, constraints will be handled via User Exits. ISPW Deploy will be enhanced in future versions so that constraints can be mapped to Categories and Environments, but for now they are used to break the Deployment Request into smaller packages so that a User Exit can modify the scheduling for an individual package.

Deployment Domains

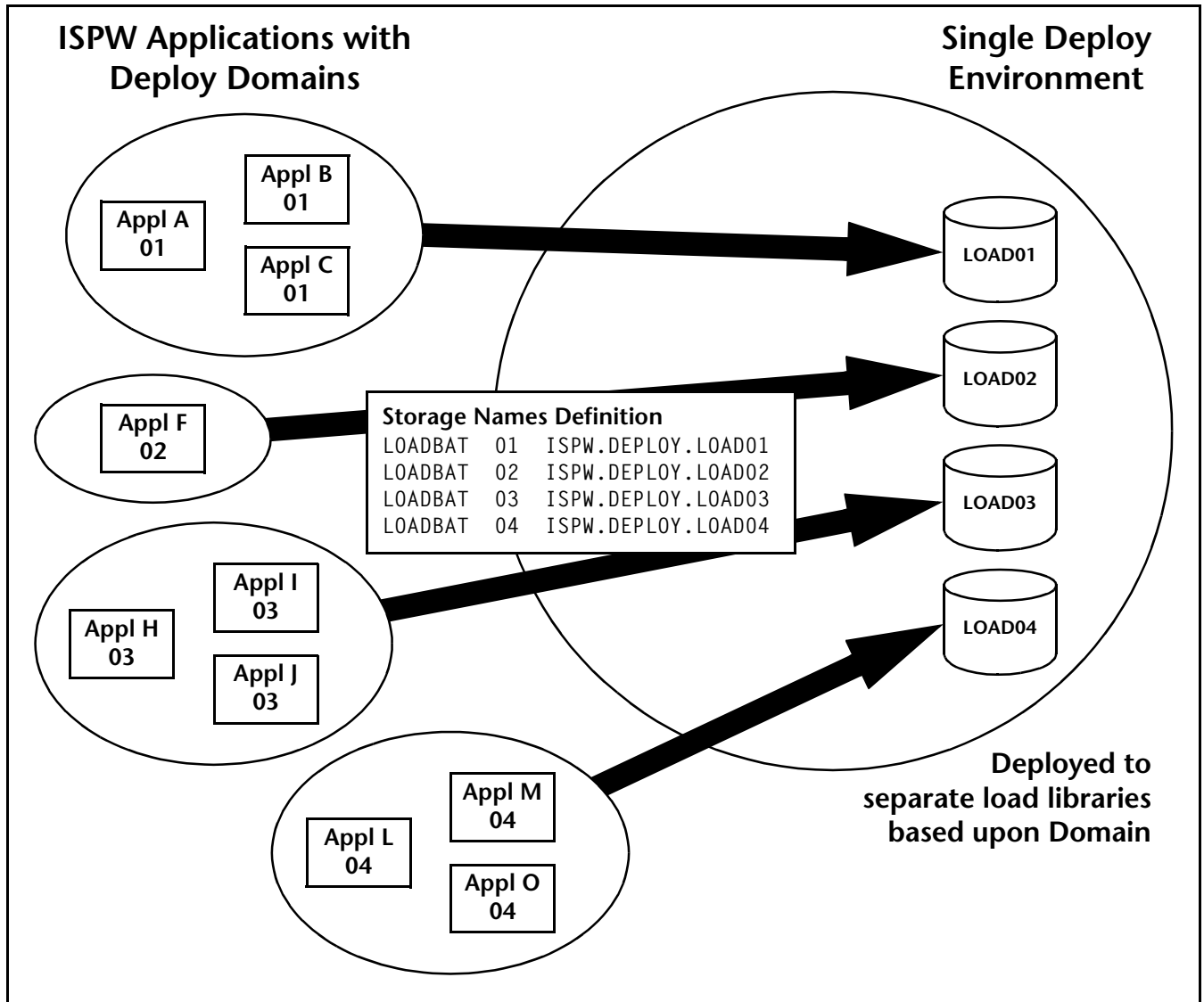
Deployment Domains provide a way to split the software components for a single Deployment Type and Sub-Environment into different storage names.

Why They are Needed

For a site with a “product” that has many thousands of components spread across multiple ISPW Applications it might not be practical to have them all deployed to a single load library (for example). By defining a Deployment Domain to each Application, it is then possible to define a different Storage Name per Deployment Domain on the target platform and so have the components spread across a number of Storage Names (i.e. datasets).

The diagram shown in [Figure 5](#) helps describe how Domains work.

Figure 5 Deployment Domains



Deploy Maintenance Functions

This chapter describes the Deploy Maintenance Functions. See [“Defining Deployment to ISPW”](#) for a guide on how to use these functions to define a Deploy Environment.

Maintenance Menu

The Deploy Maintenance Functions are found under option DP of the main ISPW Maintenance Options screen.

Figure 6 ISPW Administrator Maintenance Screen

```

ISPW M          ISPW ADMINISTRATOR MAINTENANCE (INT)
Command ==>

Reference Data ( R )          Technical Support

AD Application Streams  AN Approver Names      SM Server Maintenance
AP Applications        CH Change Codes      U User List
AR Approval Rules     CR Component Ref.   WH Warehouses
BC Base Components    ET Extension Tables WP Warehouse Policies
CC Component Category SV Servers
CT Component Types    P Prefix Numbers
EC Extension Classes  SC Set Class
ER External References GX Component groups
EX Exits              AG Application groups
ST Streams

General                    Special Technologies

X Utility Functions      T1 Technology 1
Z Generate Parms         T2 Technology 2
DP Deploy                T3 Technology 3

Enter END command to return to the previous menu.

```

Select DP to bring up the following list of Deploy Maintenance Options ([Figure 7](#)):

Figure 7 Deploy Reference Table Screen

```

ISPW M.DP       ISPW ADMINISTRATOR MAINTENANCE
Command ==>

Deploy Reference Table Maintenance

DS Systems
DD Domains
DC Categories
DT Types
DV Environments

Press END to terminate

```

More Information

The documentation in this manual is focused on the Deploy options themselves. See the *ISPW Technical Reference Guide, Chapter 3 – Maintenance Functions*, for more detailed description on how the Maintenance functions work.

DS - Deployment Systems

A Deployment System represents a physical deployment target (for example, an OS/390 LPAR). Each Deployment System must contain an ISPW CT and ISPW CI address space.

The Deploy System function lets you define a Deployment System entity.

The System entity identifies a particular system image on which many different products may be implemented. Usually this relates to some physical computer system such as an AIX or WIN server, although for operating systems like MVS it may be just a logical system image (LPAR).

On the System Table you define the System Type. Currently supported values for System Type are MVS, WIN, AIX, LINX, SUN and HPUX.



The System Type defined for a Deploy Sub-Environment must match the System Type defined for the Target System(s) in that Sub-Environment.

Figure 8 Deploy Systems Screen

```

ISPW M.R          REFERENCE DATA MAINTENANCE (INT)          UPDATE MODE
Command ==>                               Scroll==> CSR

List Commands: A Add, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, C Clone, I Import, E Export
                V View, M Modify, A Activate

  Code  Appl  Stream  Vers  Version Description  Active  Loaded  Refresh Reqd
  ___  ___  _____  ____  _____  _____  _____  _____
  DS                                0003  Deploy Import          -        -          -
  DS                                0002  New systems            Y
***** Bottom of data *****

```

Deployment System List Screen

[Figure 9](#) is a list of Deployment Systems. It is the first screen displayed after entering option DS and a version is selected.

Figure 9 Deployment System List Screen

```

ISPW          DEPLOY SYSTEM TABLE (QA)          UPDATE MODE
Command ==>                                     Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, M Modify, X Extensions

  Deploy System      System      System
  System            Type        Description
-----
- ABN-A1            MVS        Production A1
- ABN-A2            MVS        Production A2
- ABN-A3            MVS        Production A3
- ABN-FA            MVS        FA machine
- ABN-FB            MVS        FB machine - other LPAR
- ABN-FC            MVS        FC machine
- ABN-FD            MVS        FD machine
- ABN-FE            MVS        FE machine
- ABN-FG            MVS        FG machine
- ABN-T3            MVS        Test machine
- CALGARY           MVS        MVS System in CALGARY
- GABRIOLA          WIN        Gulf Islands Web Server
- GABW3H            WIN        Gabriola QA Test Server
- GABW3T            WIN        Gabriola QA Test Server
***** Bottom of data *****
    
```

Detail Screen

[Figure 10](#) is the details displayed when a specific Deployment System is selected from the Deployment System List screen or if a new Deployment System is being added.

Figure 10 Deployment System Detail Screen

```

ISPW          MODIFY DEPLOY SYSTEM DETAIL (QA)
Command ==>

Enter required details:

Deploy System (KEY) ==> CALGARY  Extensions ==>      (Y/N)
System Type           ==> MVS
Description            ==> MVS System in CALGARY

Press ENTER to complete the change or END to terminate
Note: You can add a new entry by overtyping the Key with a new unique value
    
```

Field Descriptions

[Table 3](#) describes each field for a Deployment System definition.

Table 3 Deployment System Detail

Field Name	Description
System Name	User definable system name.
Extensions	Enter a Y to go to the Extension Data definition screen. Note that on the previous screen (Figure 9) the X option can be used to navigate to the Extension Data Definition as well.
System Type	Describes the type of platform that this system represents. Values of "MVS", "AIX", "LINUX" and "WIN" are supported in this release.
Description	Free format field to hold a description of the system name.

DD - Deployment Domains

Option DD in the Deploy Maintenance Functions menu ([Figure 7](#) on page 19) displays the [Maintenance Option DD - Deployment Domains Screen](#) ([Figure 11](#)) which provides a way to split the software components for a single Deployment Type and Sub-Environment into different storage names.

Reference Data defines the external references (e.g. dataset names) used by an ISPW environment and all the applications it supports.

ISPW provides a versioning capability for the Reference Data which enables ISPW Administrators to easily implement and back out changes with the minimum disruption to operational services.

When a Reference Data maintenance dialogue is requested via option M an index of the current versions is displayed. All maintenance activities are initiated (as governed by the mode of operation UPDATE/BROWSE) using commands that are entered via this index.

Figure 11 Maintenance Option DD - Deployment Domains Screen

```

ISPW M.R          REFERENCE DATA MAINTENANCE (INT)          BROWSE MODE
Command ===>                                         Scroll===> CSR

List Commands: L Locate Entry, U Update Mode
Line Commands: S Select, V View

  Code  Appl  Stream  Vers  Version Description  Active  Loaded  Refresh Reqd
  DD
  DD                0003  ADAPT Testing          Y
  DD                0001  Doco version
***** Bottom of data *****

```

Deployment Domain List Screen

[Figure 12](#) on page 23 displays a list of Deployment Systems. It is the first screen displayed after entering option DD and a version is selected.

Reference Data defines the external references (e.g. dataset names) used by an ISPW environment and all the applications it supports.

ISPW provides a versioning capability for the Reference Data which enables ISPW Administrators to easily implement and back out changes with the minimum disruption to operational services.

When a Reference Data maintenance dialogue is requested via option M an index of the current versions is displayed. All maintenance activities are initiated (as governed by the mode of operation UPDATE/BROWSE) using commands that are entered via this index.

Figure 12 Deployment Domain List Screen

```

ISPW          DEPLOY DOMAIN TABLE (QA)          UPDATE MODE
Command ==>                                     Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, M Modify, X Extensions

  Deploy      Domain
  Domain      Description
  -----
01            Domain 01
02            Domain 02
03            Domain 03
04            Domain 04
05            Domain 05
06            Domain 06
07            Domain 07
08            Domain 08
***** Bottom of data *****
    
```

Detail Screen

[Figure 13](#) is the details displayed when a specific Deployment Domain is selected from the Deployment Domain List screen or if a new Deployment Domain is being added.

Figure 13 Deploy Domain Detail Screen

```

ISPW          MODIFY DEPLOY DOMAIN DETAIL (QA)
Command ==>

Enter required details:

  Deploy Domain (KEY) ==> 05      Extensions ==>      (Y/N)
  Description          ==> Domain 05

Press ENTER to complete the change or END to terminate
Note: You can add a new entry by overtyping the Key with a new unique value
    
```

Field Descriptions

[Table 4](#) describes each field for a Domain definition:

Table 4 Deployment Domain Detail

Field Name	Description
Domain Name	User definable Domain Name.
Extensions	Enter a Y to go to the Extension Data definition screen. Note that on the previous screen (Figure 12 on page 23) the X option can be used to navigate to the Extension Data Definition as well.
Description	Free format field to hold a description of the Domain Name.

DC – Deployment Categories

Deployment Categories are used to group Deployment Types together for activation or scheduling purposes.

Option DC in the Deploy Maintenance Functions menu ([Figure 7](#) on page 19) displays the [Maintenance Option DC - Deployment Categories Screen](#) ([Figure 14](#)) which performs maintenance of the Deployment Categories.

Figure 14 Maintenance Option DC - Deployment Categories Screen

```

ISPW M.R          REFERENCE DATA MAINTENANCE (INT)          UPDATE MODE
Command ==>                               Scroll==> CSR

List Commands: A Add, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, C Clone, I Import, E Export
                V View, M Modify, A Activate

Code  App1  Stream  Vers  Version Description  Active  Loaded  Refresh Reqd
DC
DC          0003  Deploy Import
DC          0002  Doco version          Y
DC          0001  Clone version
***** Bottom of data *****

```

Deployment Category List Screen

[Figure 15](#) displays a list of Deployment Categories. It displays when selecting a version in the [Maintenance Option DC - Deployment Categories Screen](#) ([Figure 14](#)).

The Deploy Category function lets you define a Deployment Category entity.

The Category entity allows for Deploy Types to be grouped together for deployment constraints (e.g. all types related to DB2 might be grouped together as they need to be deployed at the same time).

Figure 15 Deployment Category List Screen

```

ISPW          DEPLOY CATEGORY TABLE (QA)          UPDATE MODE
Command ==>                               Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, M Modify, X Extensions

Deploy Category      Domain
                    Description

BATCH                Batch Implementations
BATCH                PGM Batch Programs
CICS                 CICS Implementations
CICS-MQ              CICS & MQ components
DB2                  DB2 Implementations
IMS                  IMS Implementations
ONLINE               TSO Online
OPC                  OPC Scheduling
WWW                  Web Servers

***** Bottom of data *****

```

Detail Screen

[Figure 16](#) on page 25 displays the details for a specific Deployment Category you selected from the Deployment Category List screen ([Figure 15](#) on page 24) or if you add a new Deployment Category.

Figure 16 Deploy Category Detail Screen

```

ISPW          MODIFY CATEGORY DOMAIN DETAIL (QA)
Command ==>

Enter required details:

  Deploy Category (KEY) ==> CICS      Extensions ==>      (Y/N)
  Description           ==> CICS Implementations

Press ENTER to complete the change or END to terminate
Note: You can add a new entry by overtyping the Key with a new unique value
    
```

Field Descriptions

[Table 5](#) describes each field for a Category definition:

Table 5 Deployment Category Detail

Field Name	Description
Deploy Category	User definable Category Name.
Extensions	Enter a Y to go to the Extension Data definition screen. Note that on the previous screen (Figure 15 on page 24) the X option can be used to navigate to the Extension Data Definition as well.
Description	Free format field to hold a description of the Category.

DT – Deployment Types

A Deployment Type identifies a component or file type for deployment.

Option DT in the Deploy Maintenance Functions menu ([Figure 7](#) on page 19) displays the [Maintenance Option DT - Deployment Types Screen](#) ([Figure 17](#)) which performs maintenance of the Deployment Types.

Figure 17 Maintenance Option DT - Deployment Types Screen

```

ISPW M.R          REFERENCE DATA MAINTENANCE (INT)          UPDATE MODE
Command ==>                                           Scroll==> CSR

List Commands: A Add, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, C Clone, I Import, E Export
                V View, M Modify, A Activate

  Code  App1  Stream  Vers  Version Description  Active  Loaded  Refresh Reqd
  DT
  DT          0003  ADAPT Testing      Y
  DT          0002  Deploy Import
  DT          0001  Doco version
***** Bottom of data *****
    
```

Deployment Type List Screen

[Figure 18](#) on page 26 displays a list of Deployment Types. It displays when selecting a version in the [Maintenance Option DT - Deployment Types Screen](#) ([Figure 17](#)).

Figure 18 Deployment Type List Screen

```

ISPW          DEPLOY TYPE TABLE (QA)          UPDATE MODE
Command ==>                                     Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, M Modify, X Extensions

  Deploy   Deploy   Deploy Type
  Type     Category  Description

BATJCL1    BATCH      Batch System 1
DBRM       DB2         DBRM Modules
DBRMPPGM   BATCHPGM    Batch DBRM modules
DB2BIND    DB2         DB2 Bind Statements
DDL        DB2         Data Definition Language
JCL        BATCH      JCL
LISTPGM    BATCHPGM    Batch Program Listings
LOADBAT    BATCH      Batch Load Types
LOADCICS   CICS       CICS Load Types
LOADIMS    IMS        IMS Load Types
LOADPGM    BATCHPGM    Batch Programs
OBJPGM     BATCHPGM    Batch Program Objects
REXX       ONLINE     REXX Types
WEBPAGES   WWW        Static Content
***** Bottom of data *****

```

The Deploy Type function lets you define a Deployment Type entity.

The Type entity identifies a component or file type for deployment. The Deployment Type is the level at which the target files are defined, and it is also used to determine the different activation processes required for a component.

The Deployment Type corresponding to an ISPW Type/Class is specified against the Base ISPW "Stream Component Type" entry. For example, an ISPW Type of LOAD which has an ISPW Class of CICS could be specified with a Deployment Type of LOADCICS (Deployment Types can have up to 8 characters in their names).

Detail Screen

[Figure 19](#) displays the details for a specific Deployment Type you selected from the Deployment Type List screen ([Figure 18](#) on page 26) or if you add a new Deployment.

Figure 19 Deploy Type Detail Screen

```

ISPW          MODIFY DEPLOY TYPE DETAIL (QA)
Command ==>

Enter required details:

Deploy Type (KEY) ==> LOADBAT      Extensions ==>      (Y/N)

Category           ==> BATCH
Content Type       ==>              (T-Text, U-Unicode, B-Binary)
Statistics         ==>              (Y/N)
Description        ==> Batch Load Types

Press ENTER to complete the change or END to terminate
Note: You can add a new entry by overtyping the Key with a new unique value

```

Field Descriptions

[Table 6](#) describes each field for a Type definition:

Table 6 Deployment Type Detail

Field Name	Description
Deploy Type	User definable Deploy Type Name.
Extensions	Enter a Y to go to the Extension Data definition screen. Note that on the previous screen (Figure 18 on page 26) the X option can be used to navigate to the Extension Data Definition as well.
Category	Deploy Category that the Type belongs to.
Content Type	Determines the format of the Content. This is used by Deploy to ensure that the components are transferred correctly.
Statistics	Determines whether the ISPF Statistics for the PDS member are maintained. Valid values are: Y – Statistics are maintained N – Statistics are not maintained
Description	Free format field to hold a description of the Type.

DV – Deployment Environment

A Deployment Environment describes all of the definition data required for a single logical deployment.

Option DV in the Deploy Maintenance Functions menu ([Figure 7](#) on page 19) displays the [Maintenance Option DV - Deployment Environments Screen](#) ([Figure 20](#)) which performs maintenance of the Deployment Environments.

Figure 20 Maintenance Option DV - Deployment Environments Screen

```

ISPW M.R          REFERENCE DATA MAINTENANCE (INT)          UPDATE MODE
Command ==>>>                               Scroll==>> CSR

List Commands: A Add, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, C Clone, I Import, E Export
                V View, M Modify, A Activate

Code  Appl  Stream  Vers  Version Description  Active  Loaded  Refresh Req'd
DV
DV      SYSAT  0002  Import Test          Y
DV      SYSAT  0001  New version
DV      SYSET  0001  New version          Y
DV      SYSPROD 0001  New version          Y
DV      SYSST  0001  New version          Y
DV      ACTET  0001  craig et
DV      ACTST  0001  craig st             Y
DV      DON   0001  ADAPT Testing        Y
DV      ISPWPRD 0001  Initial Add
DV      ISPWQA 0001  First Cut
DV      ISPWW3M 0001  QA clone             Y
DV      ISPWW3T 0003  test refreqd        Y
***** Bottom of data *****
    
```

Environment Definition

[Figure 21](#) on page 28 displays a list of Deployment Types. It displays when selecting a Deploy Environment version in the [Maintenance Option DV - Deployment Environments Screen](#) ([Figure 20](#)).

The Deploy Environment function lets you define Deploy Environment entities.

A Deploy Environment entity describes all the definition data required for a single logical deployment. This logical Deploy Environment is associated with one or more physical Sub-Environments. Each of these Sub-Environments is a combination of various Deployment Types and Systems.

Deployment Environments are defined to ISPW against the Base ISPW "Stream level" entity. More than one Deployment Environment can be defined. For example, when software components reach "Production" they could be deployed to a "Production" environment and a "Training" environment.

Figure 21 Deployment Environment Screen

```

ISPW          DEPLOY ENVIRONMENT TABLE (QA)          UPDATE MODE
Command ==>                                     Scroll==> CSR

List Commands: B Browse Mode
Line Commands: S Sub Environments, M Modify, X Extensions

Environment   Description
ACTST        the abn deploy environment for st
*****
***** Bottom of data *****
    
```

Enter the M line command to show the [Deployment Environment Detail Screen](#) (Figure 22).

Figure 22 Deployment Environment Detail Screen

```

ISPW          MODIFY DEPLOY ENVIRONMENT DETAIL (QA)
Command ==>

Enter required details:

Environment (KEY) ==> ACTST      Extensions ==> (Y/N)
Description      ==> the abn deploy environment for st
Owner            ==> CRAIG

Press ENTER to complete the change or END to terminate
    
```

Field Descriptions

[Table 7](#) describes each field for an Environment definition:

Table 7 Deployment Environment Detail

Field Name	Description
Environment	Deploy Environment name.
Extensions	Enter a Y to go to the Extension Data definition screen. Note that on the previous screen (Figure 21 on page 28) the X option can be used to navigate to the Extension Data Definition as well.
Description	Free format field to hold a description of the Deploy Environment.
Owner	Free format field to specify the Owner of the Deploy Environment.

Sub-Environment Definition

Enter the S line command on the [Deployment Environment Screen](#) (Figure 21) to show a list of sub-environments defined for that environment in the [Deploy Sub-Environment Screen](#) (Figure 23 on page 29). It is quite normal to only have one sub-environment if that makes sense for the organization's needs.

Figure 23 Deploy Sub-Environment Screen

```

ISPW          DEPLOY SUB-ENVIRONMENT (QA)          UPDATE MODE
Command ==>                                     Scroll ==> CSR

List Commands: A Add Sub-Environment, L Locate Entry, B Browse Mode
Line Commands: T Targets, N Names, M Modify, D Delete, X Extensions

  Deploy      Sub   System Sub Environment
  Environment Env   Type   Description
-----
ACTST        CT   MVS   all types
ACTST        DB2  MVS   db2 types in this subenv
ACTST        II   MVS   just some batch stuff
ACTST        NC   MVS   rexx only
***** Bottom of data *****

```

A – Add a New Sub-Environment

To add a new sub-environment, enter **A** on the command line to display the [Add Deploy SubEnvironment Screen](#) (Figure 24).



All Targets defined for a Sub-Environment must use the same System Type as defined on this Sub-Environment table.

Figure 24 Add Deploy SubEnvironment Screen

```

ISPW          ADD DEPLOY SUB ENVIRONMENT (QA)
Command ==>

Enter required details:

Sub Environment (KEY) ==>
Description      ==>
System Type     ==>

Press ENTER to complete the change or END to terminate
Note: To add a new entry the Key must be unique

```

M – Modify Sub-Environment

Entering **M** against an existing sub-environment row presents the [Modify Deploy Sub Environment Details Screen](#) (Figure 25) where you can change the sub-environment information. The Environment information can also be changed on this screen.

Figure 25 Modify Deploy Sub Environment Details Screen

```

ISPW          MODIFY DEPLOY SUB-ENVIRONMENT DETAILS (QA)
Command ==>

Enter required details:

Deploy Sub-Environment
-----

Sub Environment (KEY) ==> CT      Extensions ==> (Y/N)
Description          ==> all types
System Type          ==> MVS

Press ENTER to complete the change or END to terminate
Note: You can add a new entry by overtyping the Key with a new unique value

```

T – Target System Selection

Selecting the Targets (Line Command T) from the [Deploy Sub-Environment Screen \(Figure 23\)](#) displays the various Target Systems that are specified and that can be changed. Each row on this [Deploy Targets Screen \(Figure 26\)](#) identifies a system to which deployment can take place for this Deploy Environment/Sub-environment combination.

Figure 26 Deploy Targets Screen

```

ISPW          ACTST - CT TARGETS          UPDATE MODE
Command ==>          Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, M Modify, X Extensions

  Deploy System      Transport Server      Staging Warehouse
  -----
  ABN-FA             WZCTW3H             WZCTW3H
  ABN-FB             WZCTW3H2            WZCTW3H2
***** Bottom of data *****
    
```

The existence of a Staging Warehouse name means that the Staging Process will take place as part of the deployment. If the Staging Warehouse is blank, then the Deployment is done directly to the target datasets according to the specified Implementation process.

Target System Detail

Select a Target System, or enter A on the command line, to display the [Target System Detail Screen \(Figure 27\)](#).

Figure 27 Target System Detail Screen

```

ISPW          MODIFY ACTST - CT TARGET
Command ==>

Enter required details:

Deploy System (KEY) ==> ABN-FA      Extensions ==> (Y/N)
Transport Server ==> WZCTW3H
Warehouse Staging? ==> Y (Y/N - If set to Y then a Warehouse of the same
                                name as the Server Name must be defined
                                in M.WH)

Press ENTER to complete the change or END to terminate
Note: You can add a new entry by overtyping the Key with a new unique value
    
```

Field Descriptions

[Table 8](#) describes each field for a Target definition.

Table 8 Target System Detail

Field Name	Description
Deploy System	System Name to which deployments are to take place. This must have been already defined in option DS – Deploy Systems.
Extensions	Enter a Y to go to the Extension Data definition screen. Note that on the previous screen (Figure 23 on page 29) the X option can be used to navigate to the Extension Data Definition as well.

Table 8 Target System Detail (*Continued*)

Field Name	Description
Transport Server	This is the name of a Component Transport Server as defined in M.SV (Servers). This relates to the physical address space/service/daemon that will undertake the Deployment on the target system.
Warehouse Staging	<p>Determines whether the Staging Phase of Deployment will be performed. Valid values are:</p> <p>Y – Staging is performed, in which case a Component Warehouse of the same name as the Transport Server needs to have been defined in M.WH.</p> <p>N – Staging is not performed</p> <p>The staging phase is optional, as depending upon whether deployment is to a remote system or not staging may or may not be preferred.</p>

N – Names

Entering **N** against an existing sub-environment row presents the [Storage Names Screen](#) ([Figure 28](#)).

Figure 28 Storage Names Screen

```

ISPW          ACTST - CT STORAGE NAMES          UPDATE MODE
Command ==>          Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, M Modify, X Extensions

  Deploy   Deploy   Staging   Storage
  Type     Domain    Type      Name

  DBRM     01        PDS       TEST.DEPLOY.ACTST.CT.DBRM
  DB2BIND  01        PDS       TEST.DEPLOY.ACTST.CT.DB2BIND
  JCL      01        PDS       TEST.DEPLOY.ACTST.CT.JCL
  JCL      06        PDS       TEST.DEPLOY.ACTST.CT.JCL
  LOADBAT  01        PDS       TEST.DEPLOY.ACTST.CT.LOADBAT
  LOADCICS 01        PDS       TEST.DEPLOY.ACTST.CT.LOADCICS
  REXX     01        PDS       TEST.DEPLOY.ACTST.CT.REXX01
***** Bottom of data *****
    
```

The Deploy Names function lets you define Logical Storage Entities.

Storage Entities describe the various attributes of the deployment targets like Storage Names and Types. You are initially presented a list of the Storage Names and Types defined for each Deployment Type/Domain.

Change a Storage Type or Name by typing over the respective values.



If a Storage Name is too large to be accommodated on the initial list screen an * is displayed alongside the Name. This is simply to warn you that you are not viewing the complete Name - If you overwrite it, the Name will be truncated.

A, S, or M – Add/Modify Names

Entering **S** or **M** against a Type or entering **A** on the command line displays the [Modify Storage Name Screen](#) ([Figure 29](#) on page 32).

Figure 29 Modify Storage Name Screen

```

ISPW          MODIFY ACTST - CT STORAGE NAME
Command ==>

Enter required details:

Environment (KEY) ==> ACTST   Sub Env (KEY) ==> CT   Extensions ==>
Type       (KEY) ==> LOADBAT Domain (KEY) ==> 01

Storage:
Use ==> R (Runtime/Temporary/Permanent)
Type ==> PDS Name ==> TEST.DEPLOY.ACTST.CT.LOADBAT
                                                    No relative path appended

Adaptation Control Cards:
Type ==> Name ==>
Implementation Process:
Type ==> S Name ==> WZR3W3H2
Job ==> WZCRAIGH Start ==> A
Activation Process:
Type ==> Name ==>
Job ==> Start ==>
Sequencing:
Group ==> Seq. Number ==> 0000 Seq. Scope ==>
Press ENTER to complete the change or END to terminate
Note: You can add a new entry by overtyping the Key with a new unique value

```

The above details define the target dataset name, implementation and Activation Processes for that Type/Domain combination.

Field Descriptions

[Table 9](#) describes each field for a Names definition.

Table 9 Deploy Storage Detail

Field Name	Description
Storage Use	<p>Deploy Storage Usage. This describes the “permanency” of the target storage name and is used to help determine whether the deployed component is available for retrieval from ISPW for browsing. The following values are possible:</p> <p>R - Runtime: The dataset is the final destination</p> <p>T - Temporary: The target is temporary and the component will not be available for retrieval.</p> <p>P - Permanent. Whilst not the final destination, it is kept here as a copy and available for retrieval.</p>
Storage Type	Type of Storage (For example: PDS, HFS)
Storage Name	Depending upon the Type specified, is the name of the target PDS or HFS Path. A storage name ending with '*' will use relative path.
Adaptation Type	<p>Type of dataset defined against Adaptation Name. Valid values are:</p> <p>SEQ – indicates that the Adaptation Name is either a sequential file or PDS with member name</p> <p>HFS – indicates that the Adaptation Name is a HFS File</p> <p>Spaces – no Adaptation to take place</p>
Adaptation Name	This is the Name of the File that holds the Adaptation Control Cards. Depending upon the Adaptation Type, it could be a HFS File (with Path), a Sequential File, PDS with member name or spaces. If spaces then ISPW will not perform any Adaptation during the Implementation phase.

Table 9 Deploy Storage Detail (*Continued*)

Field Name	Description
Implementation Process Type	<p>Defines the method by which the component part is implemented. Valid values can be:</p> <p>C – Copy. The component will be copied to the Storage Name</p> <p>S – Started Process. ISPW will initiate a Started Task that will perform the Implementation.</p> <p>X – External. A process initiated externally will drive the Implementation Process.</p>
Implementation Process Name	This is the name of a Started Task that ISPW will initiate if the Implementation Process Type is S. This can be spaces for other Process Types.
Implementation Process Job Name	This is the jobname that the Started Task will assume if the Implementation Process Type is S. This can be spaces for other Process Types.
Implementation Process Start	<p>This is the method by which the Implementation Process will be initiated</p> <p>Valid values are:</p> <p>A – Auto. The process is started automatically – valid only for a Process Type of S.</p> <p>X – External. The process is started externally – valid only for a Process Type of X.</p> <p>This value can be spaces for other Process Types.</p>
Activation Process Type	<p>Activation is an optional process that can be used to perform additional processes on the deployed component Valid values can be:</p> <p>S – Started Process. ISPW will initiate a Started Task that will perform the Implementation.</p> <p>X – External. A process initiated externally will drive the Implementation Process.</p>
Activation Process Name	This is the name of a Started Proc that ISPW will initiate if the Activation Process Type is S. This can be spaces for other Process Types.
Activation Process Job Name	This is the jobname that the Started Process will assume if the Activation Process Type is S. This can be spaces for other Process Types.
Activation Process Start	<p>This is the method by which the Activation Process will be initiated.</p> <p>Valid values are:</p> <p>A – Auto. The process is started automatically – valid only for a Process Type of S.</p> <p>X – External. The process is started externally – valid only for a Process Type of X.</p> <p>This value can be spaces for other Process Types.</p>
Sequencing Group	The sequence group name is used to define the set of packages for which processing sequence is to be controlled. Multiple sequence groups may be used in a single deployment request but the sequencing is controlled within each group independently.
Sequencing Number	This determines the sequence in which packages in a group will be processed. Multiple packages can have the same sequence number.
Sequencing Scope	<p>R Request – sequencing is applied across all target systems.</p> <p>S System – sequencing is applied within each target system.</p>

Defining Deployment to ISPW

This chapter serves as a guide to setting up Deployment, and it describes the different implementation and activation alternatives that are available.

This chapter will cover the following topics:

- [Start with a Simple Deployment](#)
- [Deployment Setup Checklist](#)
- [Using Deploy Categories](#)
- [Using Deploy Domains](#)
- [Adaptation – MVS Only](#)
- [Advanced Implementation – All Platforms](#)
- [Activation – All Platforms](#)

Start with a Simple Deployment

This section focuses on all of the steps necessary to get ISPW Deploy working for the deployment of a simple batch Cobol system to one Deploy Environment, with one Target System.

Assumptions

For a simple example, the assumptions listed in [Table 10](#) are made.

Table 10 Example Assumptions

Deployment Aspect	Assumptions
Deploy Environment	Assume that the PROD level is deployed to a separate LPAR where the production system executes.
Target System	A single Target system called PROD-SYS
ISPW Component Types	Simple system consisting of: <ul style="list-style-type: none"> • COB • LIST • COPY • LOAD • JCL • CARD (for sort control cards)
ISPW Application	A single application called PAYR in ISPW Stream BASE

More complex deploy situations (for example: Adaptation, DB2 Binding, etc.) are discussed in later sections.

1. Define a Deployment Category

A single Deployment Category is required to classify the Deploy Types. For this example, a Deploy Category of “BATCH” is sufficient. Ensure that this category is defined in the Deploy maintenance

option DC. Refer to [“Deploy Maintenance Functions” \(DC – Deployment Categories](#) on page 23) for detailed description on defining Deployment Categories.

2. Define Deploy Types

Deploy Types need to be defined in the Deploy maintenance option DT. In this simple deployment example, 3 types are required. All should be associated with the Deployment Category “BATCH” as defined in [1. Define a Deployment Category](#).

Table 11 Deploy Type Example

Deploy Type	Corresponding ISPW Type
LOADBAT	LOAD
JCL	JCL
SORTCARD	CARD

Refer to [“Deploy Maintenance Functions” \(DT – Deployment Types](#) on page 25) for detailed description on defining Deployment Types.

3. Associate Deploy Types to ISPW Types

ISPW Types defined within an Application are associated with Deployment Types at the Stream level. This is done via the Maintenance option M.ST. Select the entry, then in the Stream Definition screen, enter T to display the [Stream Base Types Screen](#) ([Figure 30](#)).

Figure 30 Stream Base Types Screen

```

ISPW M.ST/T          STREAM BASE TYPES          Row 1 of 6
Command ==>          Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, N Names, A Associations, F Flags
                X Extensions, Z Allocate

  Type  Class
S CARD
  COB
  COPY
S JCL
  LIST
S LOAD
***** Bottom of data *****

```

For this example, only types CARD, JCL and LOAD need to be selected and associated with a Deployment Type. [Figure 31](#) shows the association for type CARD.

Figure 31 Modify Stream Base Component Type Screen

```

ISPW M.ST/T          MODIFY STREAM BASE COMPONENT TYPE (INT)
Command ==>

Enter required details:

Component Type (KEY) ==> CARD          Component Domain ==>
Component Class (KEY) ==>              Deployment Type ==> SORTCARD
                                         Warehouse Populated?==>

Description ==>
*Content Type ==>
*Generate Skeleton ==>                *Test Gen. Panel ==>
*Generate Table ==>                   *Hold Gen. Panel ==>
*Generate Job ==>

*Prod. Move Job ==>                   Hold Move Time ==>           (HH:MM)

Restrict "X"? ==>                      Plibownr ==>           Plibstd ==>

Backup Lib Type ==>
Backup Lib -1 ==>
Backup Lib -2 ==>
*Model DSN(MEMBER) ==>
Alternate Type ==>
Alternate Lib ==>

Press ENTER to complete the change or END to terminate
Note: You can add a new entry by overtyping the Keys with new unique values

```

4. Define Deploy System

A single Deployment System is required for this example (PROD-SYS). See [“Deploy Maintenance Functions” \(DS - Deployment Systems](#) on page 20) for detailed description on defining Deployment Systems.

5. Define a CT Server for Remote Deployment

For Deployment to be performed on a system that is remote to the ISPW Development LPAR, an ISPW Component Transport (CT) Task needs to be defined for that remote LPAR. See the *ISPW Installation and Configuration Guide* and the current CT setup to determine the steps for doing this. Ensure that a Server definition is defined for the new server in Maintenance Option SV.

In M.SV, set Deploy Flag to "Y" for CT Task used in Deploy (either default CT Task or remote CT Task created in this step 5 or both).

For Remote Server and different LPAR: Use M.SV to add Socket number to CT Server.

6. Define a Warehouse for Staging

When deploying onto a remote LPAR, it is recommended that the components are staged locally prior to the Implementation Date. To do this, a warehouse needs to be defined for the remote CT Task in the Maintenance Option M.WH. The name of the warehouse must be the same name as the Server Name as defined in [5. Define a CT Server for Remote Deployment](#).

7. Define Deployment Domain

Deployment Domains are used when different target datasets are required for a single Deploy Type. In this simple example, it is necessary to define a Deploy Domain with a value of "01". See [“Deploy Maintenance Functions” \(DD - Deployment Domains](#) on page 22) for detailed description on defining Deployment Domains.

8. Assign Domain to Application

The Application must be associated with the Domain "01", so that when the deployment is taking place, ISPW Deploy knows that the components for this application are to be deployed into the Storage Names with a Domain of "01". Using the Maintenance option M.AD, selecting the Application/Stream version and then entering M, update the Application definition as shown in [Figure 32](#).

Figure 32 Modify Application Stream Definition Screen

```

ISPW M.AD          MODIFY APPLICATION STREAM DEFINITION (INT)
Command ==>

Enter required details:

Application        ==> PAYR
Stream             ==> BASE
Owner              ==> CRAIG
Description        ==> Payroll
Component Reference ==> Y
Deploy Domain      ==> 01

Press ENTER to complete the change or END to terminate

```

9. Define Deploy Environment

A single Deployment Environment is required for this example. The following screen shots ([Figure 33](#) through [Figure 38](#)) show how to do this.

In the Maintenance Option M.DV, enter A on the command line and press Enter to display the [Add Reference Data Version Screen](#) ([Figure 33](#)).

Figure 33 Add Reference Data Version Screen

```

ISPW M.R          ADD REFERENCE DATA VERSION (INT)
Command ==>

Enter required details:

Reference Code     ==> DV ( AD ST EC DV )
Version Description ==> First one
Reference Description ==> Production Payroll

Application Stream ( AD )      Stream ( ST )
-----
Application ==>                Stream ==>
Stream ==>                      Owner ==>
Owner ==>                       Component Ref ==>
Component Ref ==>
Deploy Domain ==>

Extension Class ( EC )      Deploy Environment ( DV )
-----
Class ==>                    Environment ==> PRODPAYR
                                Owner ==>

Press ENTER to complete the change or END to terminate

```

Enter the values as highlighted in [Figure 33](#).

Press Enter to display the [Reference Data Maintenance Screen](#) ([Figure 34](#)).

Figure 34 Reference Data Maintenance Screen

```

ISPW M.R          REFERENCE DATA MAINTENANCE (INT)          UPDATE MODE
Command ==>      Scroll==> CSR

List Commands: A Add, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, C Clone, I Import, E Export
                V View, M Modify, A Activate

   Code  N/A  DplyEnv  Vers  Version Description  Active  Loaded  Refresh Reqd
   DV    _____
S_ DV    _____ PRODPAYR 0001  First one          -      -      -
***** Bottom of data *****
    
```

Select the newly created version to present the [Deploy Environment Table Screen \(Figure 35\)](#) where you select the Environment PRODPAYR to specify the Sub-Environment Information.

Figure 35 Deploy Environment Table Screen

```

ISPW              DEPLOY ENVIRONMENT TABLE (INT)          UPDATE MODE
Command ==>      Scroll ==> CSR

List Commands: B Browse Mode
Line Commands: S Sub Environments, M Modify, X Extensions

   Environment      Description
   S PRODPAYR      Production Payroll
***** Bottom of data *****
    
```

Select the Environment PRODPAYR to display the [Deploy Sub-environments Screen \(Figure 36\)](#) where you specify the Sub-Environment Information.

Figure 36 Deploy Sub-environments Screen

```

ISPW              DEPLOY SUB-ENVIRONMENTS (INT)          UPDATE MODE
Command ==> A    Scroll ==> CSR

List Commands: A Add Sub-Environment, L Locate Entry, B Browse Mode
Line Commands: T Targets, N Names, M Modify, D Delete, X Extensions

   Deploy      Sub   System Sub Environment
   Environment  Env   Type   Description
***** Bottom of data *****
    
```

Enter the A list command to display the [Add Deploy Sub-environment Screen \(Figure 37\)](#) where you add the Sub-Environment Information.

Figure 37 Add Deploy Sub-environment Screen

```

ISPW                ADD DEPLOY SUB ENVIRONMENT (INT)                Update Complete!
Command ==>

Enter required details:

Sub Environment (KEY) ==> DFLT
Description      ==>
System Type      ==> MVS

Press ENTER to complete the change or END to terminate
Note: To add a new entry the Key must be unique

```

Specify the values as shown in [Figure 37](#) on page 40 and press Enter.

When ISPW shows the Update Complete! message, press End to return to the [Deploy Sub-environments Screen](#) ([Figure 38](#)).

Figure 38 Deploy Sub-environments Screen

```

ISPW                DEPLOY SUB-ENVIRONMENTS (INT)                UPDATE MODE
Command ==>                Scroll ==> CSR

List Commands: A Add Sub-Environment, L Locate Entry, B Browse Mode
Line Commands: T Targets, N Names, M Modify, D Delete, X Extensions

Deploy      Sub   System Sub Environment
Environment Env   Type  Description
_ PRODPAYR  DFLT MVS
***** Bottom of data *****

```

The rest of the definition is described in the following steps.

10. Add Targets to Deploy Sub-Environment

Beginning with the Deploy Environment/Sub-environment list ([Figure 39](#)):

Figure 39 Add Targets - Deploy Sub-Environment Screen

```

ISPW                DEPLOY SUB-ENVIRONMENTS (INT)                UPDATE MODE
Command ==>                Scroll ==> CSR

List Commands: A Add Sub-Environment, L Locate Entry, B Browse Mode
Line Commands: T Targets, N Names, M Modify, D Delete, X Extensions

Deploy      Sub   System Sub Environment
Environment Env   Type  Description
I PRODPAYR  DFLT MVS
***** Bottom of data *****

```

Enter T next to the Environment/Sub-environment row on the screen to display the Targets screen ([Figure 40](#)).

Figure 40 Target System Screen

```

ISPW                PRODPAYR - DFLT TARGETS                UPDATE MODE
Command ==> A                Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, M Modify,

    Deploy      Transport      Staging
    System      Server        Warehouse

***** Bottom of data *****
    
```

Enter A on the command line to display the [Add Target System Screen \(Figure 41\)](#) where you add a Target System.

Figure 41 Add Target System Screen

```

ISPW                ADD PRODPAYR - DFLT TARGET            Update Complete!
Command ==>

Enter required details:

Deploy System      (KEY) ==> PROD-SYS  Extensions ==>      (Y/N)
Transport Server   ==> WZCTW3T2
Warehouse Staging? ==> Y (Y/N - If set to Y then a Warehouse of the same
                                name as the Server Name must be defined
                                in M.WH)

Press ENTER to complete the change or END to terminate
Note: To add a new entry the Key must be unique
    
```

Enter the values as highlighted in [Figure 41](#) and press Enter.

The value “PROD-SYS” relates to the Deploy System defined in step [4. Define Deploy System](#) on page 37. The value WZCTW3T2 relates to the CT Server Name defined in step [5. Define a CT Server for Remote Deployment](#) on page 37.

Adding target systems for Distributed Servers flows an identical process – specify a Distributed Server against “Target Server”.

When ISPW shows the Update Complete! message, press End to return to the [Target System Screen \(Figure 42\)](#).

Figure 42 Target System Screen

```

ISPW                PRODPAYR - DFLT TARGETS                UPDATE MODE
Command ==> A                Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, M Modify,

    Deploy      Transport      Staging
    System      Server        Warehouse

***** Bottom of data *****
    
```

Press End to return to the Deploy Sub-Environment ([Figure 43](#)).

11. Add Deploy Type "Names"

Begin with the Deploy Sub-Environment list ([Figure 43](#)).

Figure 43 Add Deploy Type "Names" - Deploy Sub-Environment Screen

```

ISPW          DEPLOY SUB-ENVIRONMENTS (INT)          UPDATE MODE
Command ==>>                               Scroll ==>> CSR

List Commands: A Add Sub-Environment, L Locate Entry, B Browse Mode
Line Commands: T Targets, N Names, M Modify, D Delete, X Extensions

  Deploy      Sub   System Sub Environment
  Environment Env   Type   Description

  N PRODPAYR   DFLT  MVS
***** Bottom of data *****

```

Select the Names screen ([Figure 44](#)) by entering N next to the Environment/Sub-environment row on the screen.

Figure 44 Storage Names Screen

```

ISPW          PRODPAYR - DFLT STORAGE NAMES          UPDATE MODE
Command ==>> A                               Scroll==>> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, M Modify, X Extensions

  Deploy      Deploy  Storage  Storage
  Type        Domain  Type     Name
-----
***** Bottom of data *****

```

Enter A on the command line to add entries for each Deployment Type ([Figure 45](#) on page 42).

Figure 45 Add Storage Name Screen

```

ISPW          ADD PRODPAYR - DFLT STORAGE NAME
Command ==>>

Enter required details:

Environment (KEY) ==> PRODPAYR Sub Env (KEY) ==> DFLT Extensions ==>
Type (KEY) ==> LOADBAT Domain (KEY) ==> 01                               More:  +

Storage:
Use ==> R (Runtime/Temporary/Permanent)
Type ==> PDS Name ==>

Adaptation Control Cards:
Type ==> Name ==>
Implementation Process:
Type ==> C Name ==>
Job ==> Start ==> A
Activation Process:
Type ==> Name ==>
Job ==> Start ==>
Sequencing:
Group ==> Seq. Number ==> Seq. Scope ==>
Press ENTER to complete the change or END to terminate
Note: To add a new entry the Key must be unique

```

The [Add Storage Name Screen \(Figure 45\)](#) shows entries made for the “LOADBAT” deploy type to satisfy the simplest type of deploy for this type (Note the use of “01” in the domain field). See [“Deploy Maintenance Functions” \(DV – Deployment Environment on page 27\)](#) for detailed description on defining this aspect of Deployment Environments.

Entries are also needed for JCL and SORTCARD, and after the addition of these, the [Storage Names Screen](#) looks as shown in [Figure 46](#).

Figure 46 Storage Names Screen

```

ISPW          PRODPAYR - DFLT STORAGE NAMES          UPDATE MODE
Command ==>                                     Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, M Modify, X Extensions

  Deploy   Deploy   Storage   Storage
  Type     Domain   Type      Name
-----
  JCL      01       PDS       TEST.PAYR.JCL
  LOADBAT  01       PDS       TEST.PAYR.LOAD
  SORTCARD 01       PDS       TEST.PAYR.CARD
***** Bottom of data *****
    
```

Non-MVS Target Files

Specification for Deployment to non-MVS target files is identical, except that the target dataset type and name correspond to the Distributed formats as shown in [Figure 47](#).

Figure 47 Deployment to Non-MVS Target

```

ISPW          BROWSE WIN01 - DPXI STORAGE NAME
Command ==>

Enter required details:

Environment (KEY) ==> WIN01   Sub Env (KEY) ==> DPXI   Extensions ==>
Type (KEY) ==> WINEXE   Domain (KEY) ==> *                               More: +

Storage:
Use ==> R (Runtime/Temporary/Permanent)
Type ==> HFS Name ==> c\ispw\target
No relative path appended

Adaptation Control Cards:
Type ==> Name ==>

Implementation Process:
Type ==> C Name ==>
Job ==> Start ==> A

Activation Process:
Type ==> S Name ==> ACTVSCR1
Job ==> Start ==> A

Sequencing:
Group ==> Seq. Number ==> 0000 Seq. Scope ==>
Press ENTER to complete the change or END to terminate
Note: To add a new entry the Key must be unique
    
```

12. Activate Deployment Environment

The Deployment Environment is now defined and **needs to be Activated** prior to the next steps.

Enter the A line command as shown in [Figure 48](#) and press Enter.

Figure 48 Reference Data Maintenance Screen

```

ISPW M.R          REFERENCE DATA MAINTENANCE (INT)          UPDATE MODE
Command ==>                               Scroll==> CSR

List Commands: A Add, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, C Clone, I Import, E Export
                V View, M Modify, A Activate

   Code  N/A  DplyEnv  Vers  Version Description  Active  Loaded  Refresh Reqd
   DV    _____
A_ DV    _____  PRODPAYR 0001  First one          -      -      -
***** Bottom of data *****
    
```

In the [Activate Reference Data Version? Screen \(Figure 49\)](#) press Enter to activate the Deployment Environment.

Figure 49 Activate Reference Data Version? Screen

```

ISPW M.R          ACTIVATE REFERENCE DATA VERSION? (INT)    Scroll ==> CSR
Command ==>

WARNING: The following Reference Data Version will be ACTIVATED

Code  App1  Vers  Description          Active  Loaded  Refresh Reqd
  DV           0001  First one

Press ENTER to confirm the activate request or END to terminate.
    
```

With successful activation, the Active flag is set to Y as shown in [Figure 50](#).

Figure 50 Reference Data Maintenance Screen

```

ISPW M.R          REFERENCE DATA MAINTENANCE (INT)          Activate Completed
Command ==>                               Scroll==> CSR

List Commands: A Add, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, C Clone, I Import, E Export
                V View, M Modify, A Activate

   Code  N/A  DplyEnv  Vers  Version Description  Active  Loaded  Refresh Reqd
   DV    _____
  DV    _____  PRODPAYR 0001  First one          Y      -      -
***** Bottom of data *****
    
```

13. Indicate Deployment on Application

There are several places within the standard Application setup that have to be checked to make sure that the Deployment Environment just defined is actually used. Indicating that ISPW Deploy is to be called is required to be specified against the ISPW Application at the level that deployment is to take place.

Select the “PROD” level for the application in Maintenance option M.AD(L).

Figure 51 Modify Application PAYR Stream Base Screen

```

ISPW M.AD/L          MODIFY APPLICATION PAYR STREAM BASE LEVEL (INT)
Command ==>

Enter required details:

Level (KEY) ==> PROD   Promote Analysis ==> (Y - Yes)
Next Level ==>         Impact Approvals ==> (Y/S/C see help for details)
                        Impl Exit?      ==> D (D - Deploy, Y - External)

Warehouse for Sources : Name ==>         Policy ==>
                        Gen Types: Name ==> Policy ==>

Set Scheduling Information:
Set Class ==>         Job Name ==>         Queue Name ==>
Failure Notify ==>

DB2 Information:
Impl Name/Rule ==>         Name or Rule to determine Plan Implementation
DB2 Subsys ==>         Sub-system applicable for this Level
DBRM Libs ==>

XREF Name ==>         XREF Lib ==> (R or W)
Press ENTER to complete the change or END to terminate
Note: You can add a new entry by overtyping the Key with a new unique value
    
```

A **D** is required as shown in [Figure 51](#) to indicate that ISPW Deploy is to be activated for the PROD level.

14. Indicate Implementation for Each Type

Each ISPW Component Type that is participating in Deploy needs to have its “Impl Flag” set in the Maintenance option M.AD(F) ([Figure 52](#)) for the PROD level.

Figure 52 Application PAYR Stream Base Flags Screen

```

ISPW M.AD/F          APPLICATION PAYR STREAM BASE FLAGS          UPDATE MODE
Command ==>                                         Scroll==> CSR

List Commands: L Locate Entry, B Browse Mode

Type Clas  Lev      Promote  Version  Generate  Implement  Pack  Keep Build
          Method  Control  Opt Chk  Opt Chk  Src      Memb Map
-----
LOAD      HOLD
LOAD      PROD
LOAD      TEST
***** Bottom of data *****
    
```

The same should be done for the Types JCL and CARD to indicate that these types are being deployed at PROD.

15. Specify Deploy Environment

The final definition task is to associate the newly defined Deploy Environment to the ISPW Stream and Level that will be using it. This is done via the Maintenance option M.ST(L) ([Figure 53](#)) and selecting the level with an E to specify the Deploy Environment.

Figure 53 Stream Base Levels Screen

```

ISPW M.ST/L          STREAM BASE LEVELS          UPDATE MODE
Command ==>                               Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, N Names, F Flags, P Plans, X Extensions,
                Z Allocate, E Environments

Level Next Level
HOLD  PROD
E PROD
TEST  HOLD
***** Bottom of data *****

```

The [Deploy Environment Implementation Screen \(Figure 54\)](#) will be displayed where you can add the Deployment Environment.

Figure 54 Deploy Environment Implementation Screen

```

ISPW          DEPLOY ENVIRONMENT IMPLEMENTATION TABLE (INT)  UPDATE MODE
Command ==> A          Scroll==> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, M Modify

Stream      Level      Environment
***** Bottom of data *****

```

Entering an A will display the [Add Deploy Environment Implement Detail Screen \(Figure 55\)](#) on page 46).

Figure 55 Add Deploy Environment Implement Detail Screen

```

ISPW          ADD DEPLOY ENVIRONMENT IMPLEMENT DETAIL (INT)
Command ==>

Enter required details:

Stream      (KEY) ==> BASE      Extensions ==>      (Y/N)
level      (KEY) ==> PROD
Environment (KEY) ==> PRODPAYR

Press ENTER to complete the change or END to terminate
Note: To add a new entry the Key must be unique

```

The newly defined Deploy Environment “PRODPAYR” is added.

16. Refresh Server

After updating the Stream and Application details via Maintenance options M.ST and M.AD, the server will require refreshing via Maintenance option M.SM.

Deployment Setup Checklist

[Table 12](#) is a summary of the steps required to define a Deployment Environment and have it specified for an ISPW Application/Stream. It can be used as a checklist to make sure that all steps are followed.

Table 12 Deployment Setup Checklist

Step #	Step	Done
1	1. Define a Deployment Category or Categories in M.DC	
2	2. Define Deploy Types in M.DT	
3	3. Associate Deploy Types to ISPW Types in M.ST(T)	
4	4. Define Deploy System in M.DS	
5	5. Define a CT Server for Remote Deployment setup remote CT Task and define to M.SV	
6	6. Define a Warehouse for Staging with the same name as CT Task in M.WH	
7	7. Define Deployment Domain Domain in M.DD	
8	8. Assign Domain to Application in M.AD (M)	
9	9. Define Deploy Environment in M.DV including Sub-Environment.	
10	10. Add Targets to Deploy Sub-Environment in M.DV (T)	
11	11. Add Deploy Type "Names" in M.DV (N)	
12	12. Activate Deployment Environment in M.DV(A)	
13	13. Indicate Deployment on Application at the required level in M.AD(L)	
14	14. Indicate Implementation for Each Type . Set the Impl Flag for each ISPW Part that needs deploying in M.AD(F) .	
15	15. Specify Deploy Environment to Application/Stream in M.ST(E)	
16	16. Refresh Server	

Using Deploy Categories

Deployment Categories are used to group Deployment Types for activation or scheduling purposes. See [Deployment Categories](#) on page 15.

This section deals with how to use Deploy Categories within a Deployment.

Effect on Deploy Packages

Deployment requests split the deployment into separate Deploy Packages. In the simple deployment example above, Deploy Requests would have only one Deploy Package, as there is just one Sub-environment, one Deploy Target and all the components belong to the one Deploy Category.

Having a different Category for one of the Deploy Types would immediately have caused ISPW to produce two Deploy Packages for the request.

Specifying Different Categories

Using different Categories is easy. Simply define a new Category in Maintenance option DC, and associate that Category to the Deployment Type in Maintenance option DT.

Making Use of Different Categories

Specifying different Deploy Categories has the effect of creating different Deploy Packages that each contains components belonging to a single Category. These can be leveraged in the following ways:

Table 13 Deploy Category Usage

Number	Package Use	Explanation
1	Different Implementation and Activation Dates/Times	Each Deploy Package can have its own Implementation/Activation Dates and Times. The ability to set these via an exit (based upon the Deploy Category) will be made available in the next release of ISPW Deploy.
2	External Process can select Packages by Category	The External API will be enhanced in the next release of ISPW Deploy to allow the selection of all Deploy Packages for a given Category.

Using Deploy Domains

Deployment Domains provide a way to split the software components for a single Deployment Type and Sub-Environment into different storage names.

Exploiting Domains

Refer to [Deployment Domains](#) on page 16 to understand how they can be used.

Define Multiple Domains

In the Deploy Maintenance option DD it is possible to define any number of Domains. In the Simple Deployment example a Domain of "01" was defined. The Deploy Domain is free-format and any naming convention can be used.

Assign Different Domains to ISPW Applications

Deploy Domains are specified against the ISPW Application/Stream defined in Maintenance option M.AD. For the simple example above, a Domain of "01" was associated with Application PAYR. To use the power of Domains, associate different Domain Names to separate Applications.

Example

Take a situation where a site has thousands of Load modules spread across 100 ISPW Applications that need to be deployed to a set of load libraries on a target machine. By defining a Deployment Domain to each Application, it is then possible to define a different Storage Name per Deployment Domain on the target platform and thus have the components spread across a number of Storage Names (i.e. datasets).

Adaptation – MVS Only

Adaptation is a feature of ISPW Deploy which allows for the mutation of software components as they are deployed into their target environment. This feature is only available for deployments targeting MVS.

When would it be used?

Software components can be deployed to multiple systems, and in the case of JCL for example, there may be a need to change the component for environmental reasons before it is activated.

How does adaptation satisfy this requirement?

Adaptation provides a way to change the contents of a component by the use of simple change control cards, that ISPW can automatically execute per Target environment as the component is implemented.

Control Card Syntax

[Table 14](#) shows the syntax of the control cards.

Table 14 Control Card Syntax

Syntax	Description
CHG "str1" "str2"	This command replaces any occurrences of "str1" with "str2".
CCH "str1" "str2" "str3"	This command replaces any occurrences of "str1" with "str2" only when "str3" is found in the record somewhere.

The rules around the specification of these cards are listed in [Table 14](#).

Table 15 Control Card Rules

Rule #	Rule
1.	Strings must be enclosed in quotes and be delimited by a blank.
2.	Continued cards must begin with either a quote (") or blank in the first column. The command can span 3 control cards (for CHG) or 4 control cards (for CCH).
3.	No comments are possible at this time.
4.	If you want a double quote in the string, code two consecutive double quotes inside double quotes (For example: CHG "'''''' ''''').
5.	The string to be replaced in the file must be delimited on both sides by one of the following delimiters: <ul style="list-style-type: none"> • start of record • end of record • blank • ', . / < > ? " ; ^ + _ () * % =

Examples

[Table 16](#) contains some valid examples.

Table 16 Control Card Syntax and Rules Example

Control Cards	Input Records	Output Records
CHG "ONE" "AAA" CHG "TWO" "ONE"	ONE ONE ZZZ ONE TWO TWO ONE ZZZ TWO TWO	AAA AAA ZZZ AAA ONE ONE AAA ZZZ ONE ONE
CHG "ONE" "AAA" CCH "TWO" "ONE" "ZZZ"	ONE ONE ZZZ ONE TWO TWO ONE ZZZ TWO TWO	AAA AAA ZZZ AAA TWO ONE AAA ZZZ TWO TWO
CCH "TWO" "ONE" "ZZZ" CHG "ONE" "AAA"	ONE ONE ZZZ ONE TWO TWO ONE ZZZ TWO TWO	AAA AAA ZZZ AAA TWO AAA AAA ZZZ TWO TWO

Table 16 Control Card Syntax and Rules Example (*Continued*)

Control Cards	Input Records	Output Records
CHG "3380" "3390"	// UNIT=3380,VOL=SER=338010	// UNIT=3390,VOL=SER=338010 (Note: volser not changed due to delimiter rule)

Ensure Adaptation is Performed

For successful Adaptation, ensure that:

1. In M.DV under "Names" for the Deploy Type/Domain combination:
 - a. An Adaptation Type and Name are specified that hold the location of the Control Cards dataset
 - and
 - b. The Implementation Process Type must be C.
2. A dataset of the specified type and name exists on the Target that contains the Adaptation Control Cards.

[Figure 56](#) on page 50 displays a screen shot of the definition in M.DV:

Figure 56 Modify Storage Name Screen

```

ISPW          MODIFY ACTST - II STORAGE NAME
Command ==>

Enter required details:

Environment (KEY) ==> ACTST      Sub Env (KEY) ==> II      Extensions ==>
Deploy Type (KEY) ==> JCL        Domain (KEY) ==> 01
                                           More:      +

Storage:
Use ==> R (Runtime/Temporary/Permanent)
Type ==> PDS Name ==> TEST.DEPLOY.ACTST.II.JCL
                                           No relative path appended

Adaptation Control Cards:
Type ==> SEQ Name ==> TEST.DEPLOY.ADAPT.CARDS(TEST01)

Implementation Process:
Type ==> C Name ==>
Job ==> Start ==> A

Activation Process:
Type ==> Name ==>
Job ==> Start ==>

Sequencing (not currently used):
Group ==> Seq. Number ==> 0000 Seq. Scope ==>
Press ENTER to complete the change or END to terminate
Note: You can add a new entry by overtyping the Key with a new unique value

```

The member TEST01 in PDS dataset "TEST.DEPLOY.ADAPT.CARDS" contains the Control Cards and must exist on the Target platform.

Advanced Implementation – All Platforms

This "Simple Deployment" described at the beginning of this chapter, uses an Implementation Process of C to effect a copy to the target dataset. This section describes the more advanced Implementation Processes.

Implementation Processes

[Table 17](#) lists the three Process Types.

Table 17 Implementation Process Types

Type	Description
C	<p>Copy: The component will be copied to the Target.</p> <p>When the Implementation Time is reached, the Target CT or Remote Server takes the component from the staging warehouse (if defined) and copies it to the Target Storage Name. If an Adaptation dataset is specified (for MVS only) then the Target CT Server will perform the Adaptation as it copies it.</p>
S	<p>Started Process: ISPW will initiate a Process to perform the implementation process.</p> <p>When the Implementation Time is reached, the Target CT or Remote Server starts the named command. For MVS this will be a Started Task and if a job name is specified then this is passed as a parameter.</p> <p>For Distributed Platforms, Processes started by the Remote Server may be programs, or PHP scripts. The deploy package and environment details will be passed as command line arguments to the started process.</p> <p>The Process responsible for the Implementation Process and must make Deploy API Calls to retrieve the components itself and implement them. See "Deployment Processes" for more information on how this is done.</p>
X	<p>External: A process external to ISPW will perform the implementation process.</p> <p>For Implementation Processes defined as External, ISPW Deploy will do nothing when the Implementation Time is reached. It is the responsibility of an external process to make the necessary API Calls to retrieve the components and implement them. The external process is started outside of ISPW Deploy (For example: by a Scheduler).</p>

Activation – All Platforms

Activation is an optional separate process that comes after the Implementation process. It is useful when the deployment of a component type involves multiple steps that need to be reflected in ISPW.

Example of Use

The deployment of DDL could be split into Implementation and Activation processes. The Implementation process could copy the components into an external dataset, and then the Activation process could be a bulk process that was externally initiated to process the DDL. A call at the end of the process uses the ISPW API to indicate that Activation was complete.

Activation Processes

[Table 17](#) lists the 2 possible process types.

Table 18 Process Types

Type	Description
S	<p>Started Process: ISPW will initiate a Process to perform the implementation process.</p> <p>When the Activation Time is reached, the Target CT or Remote Server starts the named command. For MVS this will be a Started Task and if a job name is specified then this is passed as a parameter.</p> <p>For Distributed Platforms, Processes started by the Remote Server may be programs, batch commands or shell scripts, or PHP scripts. The deploy package and environment details will be passed as command line arguments to the started process.</p> <p>The PROC is responsible for the Activation Process and must make Deploy API Calls to retrieve the components itself and activate them. See "Deployment Processes" for more information on how this is done.</p>

Table 18 Process Types (*Continued*)

Type	Description
X	External: A process external to ISPW will perform the implementation process. For Activation Processes defined as External, ISPW Deploy will do nothing when the Activation Time is reached. It is the responsibility of an external process to make the necessary API Calls to retrieve the components and activate them. The external process is started outside of ISPW Deploy (For example: by a Scheduler).

Deployment Processes

Overview

Deployment processes extend the capability of ISPW Deploy by enabling you to build your own processes on the target platform for implementation and activation.

ISPW Standard Implementation

ISPW Deploy provides a standard “COPY” process that implements a package of component parts into a target dataset (PDS, HFS, Unix or Windows directory) at a particular date and time. This is sufficient for the deployment of components that do not need any processing around their implementation.

More Complex Implementation Requirements

The implementation of many types of component parts is more complex than the standard copy process can satisfy. Examples include:

- DB2 components (for example, DDL, DBRMs)
- J2EE components (for example, .ear files)
- Natural
- Etc.

Deployment Processes

Deployment Processes provides a mechanism for customers to be able to write their own implementation processes. This is done by providing ISPW Service calls that are called via an interface native for the target platform to perform the following functions:

- Start and End the processing for a Physical Package
- Retrieve individual component items from ISPW that belong to the Package
- Indicate to ISPW Deploy that the processing for a component part has finished
- Retrieve a list of Packages ready for Deployment

Starting a User Process

User Processes can be started in two ways:

- Automatically by ISPW at Implementation or Activation Time
- By some process external to ISPW

ISPW Started Process

The steps listed in [Table 19](#) occur for an ISPW initiated Deployment Process.

Table 19 ISPW Initiated Deployment Process

Step #	Process
1.	ISPW CM communicates to the ISPW CT/Remote Server on the Target System to start a Proc for a specific Deploy Package.

Table 19 ISPW Initiated Deployment Process (*Continued*)

Step #	Process
2.	<p>The CT/Remote Server on the Target System initiates User Process. Depending upon the Target Platform the following command is issued: An MVS based CT issues:</p> <p>START <i>procname</i>,JOBNAME=<i>jobname</i>,DREQID=<i>ReqID</i>,PKGID=<i>PkgID</i>,SYSTNAME=<i>SystemName</i></p> <p>Unix and Windows based Remote Servers spawn a process that can be either a program, batch command or shell script, or PHP script. The only constraint being that it must exist in the directory specified by the COMMAND_PATH variable in the rparms.ini file.</p>
3.	User Process starts and communicates with the ISPW CM Task to indicate that it is ready to process the Physical Package. ISPW CM passes back the list of Items to process and updates the status of the Package to "Executing".
4.	The User Process communicates with the ISPW CM Task to retrieve a Component Part from the Staging Warehouse. The CM Task co-ordinates with the Target CT Task to copy the Component Part to the Target Dataset.
5.	The User Process performs Processing on the Component Part (or Parts) as required and keeps ISPW up-to-date with the status of each Item in the Package.
6.	The User Process will communicate to the ISPW CM Task when all processing is complete for a Package. The CM Task will update statuses accordingly.

Deploy Process for MVS

This section describes the environment that an MVS process runs under and a description of the calls available on the MVS Platform.

Process Start

The CT address space starts an MVS Started Process – in the Samplib is an ISPWRX member containing the JCL as shown in [Figure 57](#).

Figure 57 JCL in Member ISPWRX

```
//ISPWRX  PROC DREQID=MISSING,PKGID=MISSING,SYSTNAME=MISSING
//*
//*   ISPW DEPLOY IMPLEMENTATION PROCESSOR
//*
//WZZRX   EXEC PGM=IKJEFT01,REGION=0M,TIME=1440,DYNAMNBR=99,
//   PARM='%WZURX### &DREQID &PKGID &SYSTNAME WZU@ACI# ISPF (DEBUG(NO) '
//*
//STEPLIB DD DISP=SHR,DSN=ISPW.BASE.LOAD
..
..
..
..
```

This process runs TSO and then initiates a REXX, passing the parameters that identify the Physical Package. An ISPW CI Task must be present on the system to facilitate communication with the CM Task (which can be on a remote LPAR).

Passed Parameters

The three parameters listed in [Table 20](#) are passed from ISPW to the Started Proc and then to the REXX. These uniquely identify the Physical Package.

Table 20 Passed Parameters

Parm	Description
DREQID	Deploy RequestID
PKGID	PackagelD
SYSTNAME	System Name

Initial REXX

The initial REXX parses the passed parameters and then calls the ISPW Program WZZRX. This program creates an ISPW Environment within which the API Calls can be made. WZZRX is passed the three parameters that the Proc was passed plus the name of the Implementation REXX - WZU@ACI# in the example and supplied in the Samplib.

WZU@ACI#

The REXX makes calls to ISPW to enable it to perform the Implementation and Activation Processes. The sample in the Samplib contains example processes for different deployment types. The next section describes the standard Deploy API calls that the deployment REXX would use.

MVS API Call – PACKAGE START

This initiates the execution of a package and returns the package details and complete list of items for the REXX to start processing.

Syntax

```
WZZRXI("PACKAGE", "START", parm1, parm2, parm3, parm4, parm5)
```

Input Parameters

[Table 21](#) lists the input parameters for the API call.

Table 21 Input Parameters - PACKAGE START

Parm	Description
RequestID	This is one of the passed parameters.
PackagelD	This is one of the passed parameters.
System Name	This is one of the passed parameters.
Package Stem	The name of a REXX Stem variable that will be populated with the returned Package variable names and values. This is a user supplied name
Item Stem	The name of a REXX Stem variable that will be populated with the returned Item variable names and values. This is a user supplied name

Returned Values

[Table 22](#) lists the returned values that are stored in REXX Stem variables, and consist of a single set of Package variables and n sets of Item variables, where n is the number of Items in the Package.

Table 22 Returned Values - PACKAGE START

Return Variables	Description
<i>PkgStem.dreqid</i>	RequestID.
<i>PkgStem.pkgid</i>	PackageID.
<i>PkgStem.systname</i>	System Name.
<i>PkgStem.ctsrvrnm</i>	The name of the ISPW CT Server on the target machine.
<i>PkgStem.dpenv</i>	Deploy Environment.
<i>PkgStem.subenv</i>	Deploy Sub-environment.
<i>PkgStem.dpcat</i>	Deploy Category.
<i>PkgStem.process</i>	Current Deploy Process; value can be: S – Staging I – Implementing A – Activating
<i>PkgStem.procstat</i>	Current Process Status: value should always be E (for executing).
<i>PkgStem.erlydate</i>	Earliest Start Date for the Process.
<i>PkgStem.erlytime</i>	Earliest Start Time for the Process.
<i>PkgStem.strtdate</i>	Actual Start Date for the Process.
<i>PkgStem.strttime</i>	Actual Start Time for the Process.
<i>ItemStem.0</i>	This contains the number of returned items. There is a set of returned values for each Item returned, and they are identified by the suffix “.n” in the stem name.
<i>ItemStem.dreqid.n</i>	RequestID.
<i>ItemStem.pkgid.n</i>	PackageID.
<i>ItemStem.itemid.n</i>	Item Number.
<i>ItemStem.systname.n</i>	System Name.
<i>ItemStem.status.n</i>	Physical Item Status.
<i>ItemStem.applid.n</i>	ISPW ApplicationID.
<i>ItemStem.stream.n</i>	ISPW StreamID.
<i>ItemStem.curlevel.n</i>	ISPW Level at which the component is at.
<i>ItemStem.cmpntype.n</i>	ISPW Component Type.
<i>ItemStem.cmpnclas.n</i>	ISPW Component Type Class.
<i>ItemStem.partname.n</i>	Name of the Item.
<i>ItemStem.parttype.n</i>	ISPW Part Type
<i>ItemStem.partclass.n</i>	ISPW Part Class
<i>ItemStem.dptype.n</i>	Deploy Type
<i>ItemStem.action.n</i>	ISPW action code. Value can be: B lank – normal D – Delete F – Fallback I – Implement. The object does exist in a higher level, and it is necessary to reinstate the runtime environment to what it was previously. To do this, it is necessary to Implement the previous object in the runtime environment. No action is required; the previous object should be restored to the runtime environment.

Table 22 Returned Values - PACKAGE START (*Continued*)

Return Variables	Description
<i>ItemStem.stortype.n</i>	Storage Type defined for this Item – goes with Storage Name. Value can be: PDS – Partitioned Dataset HFS – Unix File system
<i>ItemStem.storname.n</i>	Name of the Target Dataset defined in Deploy.
<i>ItemStem.storuse.n</i>	Storage Use. Indicates the “permanency” of the component part in the Storage Name. Value can be: R – Runtime T – Temporary P – Permanent
<i>ItemStem.altstype.n</i>	Not currently used.
<i>ItemStem.altname.n</i>	Not currently used.

The “.n” suffix on the ItemStem variable names represents the instance of that variable in the array. The first item will have variable names of the format:

```
ItemStem.VariableName.1
```

Subsequent items will be .2 .3 etc.

The number of Items is returned in ItemStem.0.

Return Codes

[Table 23](#) lists the possible Return Codes.

Table 23 Return Codes - PACKAGE START

Value	Description
0	Successful
8	Failure
20	Parameter error

Example

The code shown in [Figure 58](#) on page 58 makes a PACKAGE START call and then writes the values out for some of the variables.

Figure 58 PACKAGE START Call

```

RC = WZZRXI("PACKAGE", "START", INDREQID, INPKGID, INSYSTNAME, "PKG", "ITEM")
SAY "RC=" RC
IF RC <> 0 THEN DO
  SAY "ERROR PACKAGE START " RC
  SAY ERRTXT
  EXIT RC
END
SAY "GOT " ITEM.0 " RETURNED ITEMS"
SAY "PKG.DREQID=" PKG.DREQID
SAY "PKG.PKGID=" PKG.PKGID
SAY "PKG.SYSTNAME=" PKG.SYSTNAME
..
..
..
DO I = 1 TO ITEM.0 BY 1
  SAY "ITEM " I
  SAY "ITEM.DREQID=" ITEM.DREQID.I
  SAY "ITEM.PKGID=" ITEM.PKGID.I
  ..
  ..
END

```

MVS API Call – ITEM GET

This call causes ISPW to copy the Component Part from the Staging Warehouse into a specified Dataset.

Syntax

```
WZZRXI("ITEM", "GET", parm1, parm2, parm3, parm4, parm5)
```

Input Parameters

[Table 24](#) lists the input parameters for the API call.

Table 24 Input Parameters - ITEM GET

Parm	Description
RequestID	Identifies the Deploy Request
PackageID	Identifies the Package in the Deploy Request.
ItemID	This uniquely identifies a component part within the Package.
System	System Name.
Target Dataset	The name of a Target Dataset where Deploy is to copy the component part to.

Returned Values

For non-zero return codes, an error message is returned as follows, otherwise no values are returned. The returned values listed in [Table 25](#) are stored in REXX variables.

Table 25 Returned Values - ITEM GET

Return Variables	Description
Errid	ErrorID.
errtext	Error Text.

Return Codes

[Table 26](#) lists possible Return Codes.

Table 26 Return Codes - ITEM GET

Value	Description
0	Successful
8	Failure
20	Parameter error

Example

The code shown in [Figure 59](#) makes an ITEM GET call for every Item returned from the Package Start. Each Part is copied to a PDS.

Figure 59 ITEM GET Call

```
TDSN = 'ISPW.DEPLOY.IMPL'
DO I = 1 TO ITEM.0 BY 1
  RC = WZRRI("ITEM","GET",ITEM.DREQID.I,ITEM.PKGID.I,,
            ITEM.ITEMID.I, ITEM.SYSTNAME.I, ,
            TDSN("||ITEM.PARTNAME.I||"))
  IF RC <> 0 THEN DO
    SAY "ERROR ITEM GET " RC
    SAY ERRID ERRTXT
    EXIT RC
  END
END
```

MVS API Call – ITEM END

This call is to tell ISPW that the processing for an Item is finished.

Syntax

```
WZRRI("ITEM","END",parm1,parm2,parm3,parm4,parm5[,parm6])
```

Input Parameters

[Table 27](#) lists the input parameters for the API call.

Table 27 Input Parameters - ITEM END

Parm	Description
RequestID	Identifies the Deploy Request
PackageID	Identifies the Package in the Deploy Request.
ItemID	This uniquely identifies a component part within the Package.
System	System Name.
Completion Status	This is the completion status. Allowed values are: C – Complete F – Failed
Error Text	This is an optional parameter that is used to supply an error message to ISPW in the event of a Failed successful status. If no message is sent for a Failed status an internal message of "Item Marked not complete" is set.

Returned Values

For non-zero return codes, an error message is returned as follows, otherwise no values are returned. The returned values listed in [Table 28](#) are stored in REXX variables.

Table 28 Returned Values - ITEM END

Return Variables	Description
Errid	ErrorID.
errtext	Error Text.

Return Codes

[Table 29](#) lists possible Return Codes.

Table 29 Return Codes - ITEM END

Value	Description
0	Successful
8	Failure
20	Parameter error

Example

The code shown in [Figure 60](#) makes a ITEM END call for every Item returned from the Package Start, and marks each one as failed.

Figure 60 ITEM END Call

```
DO I = 1 TO ITEM.0 BY 1
  RC = WZZRXI("ITEM","END",ITEM.DREQID.I,ITEM.PKGID.I,,
             ITEM.ITEMID.I, ITEM.SYSTNAME.I, "F", "FAILED ON PURPOSE")
  SAY "RC=" RC
  IF RC <> 0 THEN DO
    SAY "ERROR ITEM END " RC
    SAY ERRID ERRTXT
    EXIT RC
  END
END
```

MVS API Call – PACKAGE END

This call communicates to ISPW that processing has ended for this Package.

Syntax

```
WZZRXI("PACKAGE", "END", parm1, parm2, parm3, parm4, [parm5])
```

Input Parameters

[Table 30](#) lists the input parameters for the API call.

Table 30 Input Parameters - PACKAGE END

Parm	Description
RequestID	Identifies the Deploy Request
PackageID	Identifies the Package in the Deploy Request.
System	System Name.

Table 30 Input Parameters - PACKAGE END (*Continued*)

Parm	Description
Completion Status	This is the completion status. Allowed values are: C – Complete F – Failed
Error Text	This is an optional parameter that is used to supply an error message to ISPW in the event of a Failed successful status. If no message is sent for a Failed status an internal message of “Package Marked not complete” is set.

Returned Values

For non-zero return codes, an error message is returned as follows, otherwise no values are returned. returned values listed in [Table 31](#) are stored in REXX variables

Table 31 Returned Values - ITEM END

Return Variables	Description
Errid	ErrorID.
errtext	Error Text.

Return Codes

[Table 32](#) lists possible Return Codes.

Table 32 Return Codes - ITEM END

Value	Description
0	Successful
8	Failure
20	Parameter error

Example

The code shown in [Figure 61](#) makes a PACKAGE END call.

Figure 61 PACKAGE END Call

```
Package_Error_Msg="Job "Jobname jnum " Failed"
wrc = WZRRI("PACKAGE","END",.
  Request_ID,Package_ID,Target_System,"F",Package_Error_Msg)
```

MVS Deploy Log

The Deploy Log provides a way for the results of a deployment process to be written to a local dataset which can then be retrieved within the ISPW User Interface. This provides error handling back to a “Single Point of Control” even for deployments to LPARs with no shared DASD.

Log Dataset

M.ER variables RXDPLOGP and RXDPLOGA determine the name and allocation parameters. In the WZU@ACI# samplib the code shown in [Figure 62](#) on page 62 is used to allocate the dataset.

Figure 62 Allocate DP Log

```

Allocate_DP_Log:
/* ----- */
/* Allocate the Deploy Log which is viewable from the DP Package list */
/* in ISPW. */
/* ----- */

filetype = "RXDPLOGA"
"ISPEXEC TBGET $EXTABLE"
If rc = 0 ,
Then rxlogalcl = extlfile
Else rxlogalcl = "SP(1 1) TRACKS"

filetype = "RXDPLOGP"
"ISPEXEC TBGET $EXTABLE"
If rc = 0 ,
Then rxlogpfx = extlfile
Else rxlogpfx = SYSVAR("SYSPREF")
rxlogdsn=""rxlogpfx||".REQ"||Request_ID||".PKG"||,
Package_ID||".#"||Strip(Left(Target_System,7))||".LOG"
If Sysdsn(rxlogdsn) = "DATASET NOT FOUND" Then
Do
Address TSO "ALLOC F(WZZDPLOG) DA("rxlogdsn") " rxlogalcl ,
"LRECL(137) BLKSIZE(4000) RECFM(V B) NEW CATALOG"
If rc = 0 Then
Do
Address TSO EXECIO 0 "DISKW WZZDPLOG (OPEN FINIS"
Address TSO "FREE F(WZZDPLOG)"
End
End
Address TSO "ALLOC F(WZZDPLOG) DA("rxlogdsn") SHR"

rc=WZZUTI("SETLOG", "WZU@ACI#", "DD:WZZDPLOG", log)

Return

```

Logging to the Dataset

Within the Deployment process, a REXX function is used to log messages to the dataset. There are examples of this in the sample exit (WZU@ACI#), and [Figure 63](#) shows a snippet of sample code.

Figure 63 Log Messages Sample Exit

```

If Item_rc<>0 Then Do
Item_Error_Msg = 'IEBCOPY error writing to:' mfstarg
log = Item_Error_Msg
rc=WZZUTI("SETLOG", "WZU@ACI#", "DD:WZZDPLOG", log)
Say Item_error_Msg
End
Else Do
log = logmem||':Successful load of members to:' mfstarg
rc=WZZUTI("SETLOG", "WZU@ACI#", "DD:WZZDPLOG", log)
End

```

Keeping the Logs

It is recommended that the logs be maintained for a reasonable period of time, for example, months, so that any deploy issues can be resolved.

Viewing the Logs

The logs can be viewed from the Deploy Package screen within the Deploy Request off main menu option DP.

Deployment Process for Distributed Platforms

This section describes options for the Implementation and Activation phases of deploy packages on Windows, Linux, and Unix systems.

Distributed Deploy Definition Concepts

Implementation – Basic Copy

A basic deployment of a package will usually stage the package files to a warehouse on the target system, then implement them by copying them from the warehouse to the target location. ISPW provides options to support more complex deployment operations.

Pre and Post Exits for Basic Copy

It is possible to start a process before and/or after the basic package implementation copy. It is also possible to start a process before and/or after each file copy. These processes could check space availability in the target location, stop applications or services, or any other function. If the exit process returns an error, the deploy request is failed. See [Table 33](#) for details on the type of exit processes that can be used.

Implementation – Advanced Process

If the deploy requires more control over the implementation (than just a copy), an implementation process can be specified. In this case, the implementation process will get started by the Remote Server at the beginning of the implementation phase. It can retrieve package details, run the file copies, and perform any additional operations. When using this option, no pre/post exits will be run. This option should only be used when one of the other options will not achieve the required results, as it puts the complete deploy package responsibility in the started implementation process. See [Table 33](#) for details on the type of implementation processes that can be used.

Activation Process

Following an implementation, a process may be started to perform additional operations, such as restarting an updated application. The activation process will be started by the Remote Server on the target system. The process may be a program, a batch file or shell script, or a PHP script. See [Table 33](#) for details on the type of activation processes that can be used.

Start Process Options

The Remote Server will start processes during the deploy package phases, based on the definitions for the deploy environment and sub-environment. Processes can be started at the points as listed in [Table 33](#), based upon the definitions in M.DV(N).

Table 33 Start Process Options

Description	Implement	Activate	Pkg-Pre	Pkg-Post	Item-Pre	Item-Post
Implementation is X or blank	N	N	N	N	N	N
Implementation is C, Activation is X or blank	N	N	Y	Y	Y	Y
Implementation is C, Activation is S	N	Y	Y	Y	Y	Y
Implementation is S, Activation is X or blank	Y	N	N	N	N	N
Implementation is S, Activation is S	Y	Y	N	N	N	N



Pre and Post exists are only run if they are defined in the WRSDPEX extension class for the Deploy Environment or Sub-Environment. If the exits are defined and can be run based on the above chart, then the Remote Server will start them.

WRSDPEX

If pre or post exits are to be executed, then the WRSDPEX Extension Class must be defined in M.EC and then used against the Deploy Environment defined in M.DV. The definition is shown in [Figure 64](#).

Figure 64 WRSDPEX Extension Class

Name	Type	Length	Description
ITEMPOST	CHAR	8	Item Post-Implementation Copy Exit
ITEMPRE	CHAR	8	Item Pre-Implementation Copy Exit
PKGPOST	CHAR	8	Package Post-Implementation Copy Exit
PKGPRES	CHAR	8	Package Pre-Implementation Copy Exit

Processes

Processes started by the Remote Server may be programs, batch commands or shell scripts, or PHP scripts. The deploy package and environment details will be passed as command line arguments to the started process. If the process is a pre/post exit, then only a return code is expected to be returned. If the process is for package activation, then the status of the activation should be set by the process.

Process Name

The name of the deploy process to be run is an 8 character name defined in the deploy environment in M.DV. This name is used to locate the command file in the directory specified by the "COMMAND_PATH" value in the Remote Server start up parameters.

On a Windows system, the process name is appended with a ".exe", then a ".bat", then a ".php" to locate a file that matches. If a file matches the name, using a case insensitive check, then that file is used to start the deploy process. If no match is found, then the deploy process is failed.

On a Linux or Unix system, a matching file is checked for, then the process name is appended with a ".sh", then a ".php" to locate a file that matches. Each check is performed once in upper case, and once in lower case. If a file matches the name, then that file is used to start the deploy process. If no match is found, then the deploy process is failed.

Programmatic Considerations

When deciding whether to use a program, batch/shell script, or PHP script, you will need to consider what the process will be doing. By far the best choice for these started processes is a PHP script. ISPW provides a set of PHP functions that can be used in your scripts to communicate directly with the ISPW server and perform other useful operations. There are also a set of logging functions that will ensure that messages generated by the script will be placed in the deploy log and can be viewed from the ISPW Client. PHP is freely available on all platforms supported by ISPW. ISPW Support can provide PHP install instructions and the ISPW PHP Scripting package.

In some cases, specialized utility programs will be provided for use as the implementation or activation process.

If you do not have PHP available and do not want to install it, you may use batch commands or shell scripts to perform activation processing. These bat/shell scripts do not have a direct communication interface to the ISPW server, but a command line utility is provided to let the script place entries into the deploy process log that will be picked up by the Remote Server so it can report package status

back to the ISPW server. You may also use bat/shell scripts for pre/post exits, as all that is expected from these exits is a return code. Using a bat/shell script for the implementation process is likely not possible, due to the requirement for direct communications with the ISPW server.

PHP Considerations

The best option for deploy processes is to use the ISPW PHP Scripting package. The package contains the PHP function libraries, the ISPW interface utility, and documentation on usage in HTML format. For the latest version of the ISPW PHP Scripting package, contact Compuware Customer Solutions.

PHP scripts may be used for the activation processing, implementation processing, and implementation pre/post exits.

PHP Example

The code shown in [Figure 65](#) on page 66 is an example of an implementation process using PHP.

Figure 65 Sample Implementation Process Using PHP

```

<?php
/*
 * Sample Implementation Script
 */
require("wzapi.php");
global $wzenv;

/* Initialize ISPW Environment */
WZAPI($argv,NULL,NULL);

WZLogInit("append,info");
WZLogMsg("MSG", "Deploy Implementation Started");

/* Initialize connection to ISPW */
$srid = WZGetArgvValue($argv,"srid");
$rc = WZInit($srid);
if ($rc == -1) {
    WZLogMsg("ERROR", "ISPW INIT Failed: ".WZLastError() );
    return(8);
}

/* Get Command Line Arguments */

$dreqid = WZGetArgvValue($argv,"dreqid");
$pkgid = WZGetArgvValue($argv,"pkgid");
$systname = WZGetArgvValue($argv,"systname");

$pkgarray = WZPackageStart($dreqid,$pkgid,$systname);
if (!$pkgarray) {
    WZLogMsg("ERROR", "Package Start Failed: ".WZLastError() );
    return(8);
}

/* Process the Deploy Package */

$pk = $pkgarray['PK'];
WZLogArray("INFO","PK Data Group",$pk);

/* Process each Item in the Package */

$FailCount=0;
foreach ($pkgarray['ITEMS'] as $id=>$item)
{
    WZLogArray("INFO","ITEM Data Group",$item);
    if ($item['status'] != "C")
    { /* Use Random Number to determine whether we will process this item */
        $random = rand(1, 11);
        if ($random > 5)
        { /* If random number is greater than 5 then fail the Item */
            $rc = WZItemFail($item,"Package ". $item['pkgid'] . " Item ". $item['itemid'] .
                " Failed - Random Number=" . $random);
            $FailCount=$FailCount+1;
        }
        else
        {
            $item['targtype'] = $item['stortype'];
            $item['targname'] = trim($item['storname']) . $wzenv['sc'] .
                trim($item['cmpntype']) . $wzenv['sc'] .
                trim($item['cmpnname']) . ".txt" ;

            $itemr = WZItemGet($item);
            if ($itemr)
            {
                $rc = WZItemComplete($item);
                $msgbuf = "Deploy Copy to " . $item['targname'] . " Complete";
                WZLogMsg("MSG", $msgbuf);
            }
            else
            { /* ItemGet Failed */
                $rc = WZItemFail($item,"Package ". $item['pkgid'] . " Item ".
                    $item['itemid'] . " Failed - WZItemGet RC=" . WZLastRC());
                $FailCount=$FailCount+1;
            }
        }
    }
}

/* Fail the package if any items failed , otherwise Complete the Package */

if ($FailCount > 0)
    $rc = WZPackageFail($pk,"Package " . $pkgid . " failed because " .
        $FailCount . " item(s) failed");
else
    $rc = WZPackageComplete($pk);

WZLogMsg("MSG", "Deploy Implementation Ended");
WZTerm();
exit(0);
?>

```

Windows Batch Commands

You may use Windows batch commands to perform deploy package activation processing and pre/post exits. You may not use batch commands for deploy implementation processing.

Batch commands have no direct access to the ISPW server, and will need to report the status of the activation by adding entries to the deploy package log file. The WZUDPAP command line utility is used to create the log, add entries to it, and report activation status. Refer to [WZUDPAP – Writing to the Log](#) on page 70 for details.

When using batch commands for activation processing, you will need to make use of the Deploy Monitor feature of the Remote Server. Refer to [Deploy Activation Monitor](#) on page 68 for details.

Windows Batch Command Example

[Figure 66](#) shows a sample Windows batch command for Package activation.

Figure 66 Windows Batch Command

```
REM Sample Deploy Activation Command
REM Syntax: "sample.bat DREQID=nnn PKGID=nn SYSTNAME=xxx LOGPATH=111 SRID=sss"
REM Signal that activation has started
wzudpap.exe START %1=%2 %3=%4 %5=%6 %7=%8 %9=%10
REM Do activation stuff here. Set "RC" to return code
wzudpap.exe MSG %1=%2 %3=%4 %5=%6 %7=%8 TEXT="This message added to deploy log"
IF %RC% GTR 0 GOTO :Failed
:Complete
REM Signal that activation has completed
wzudpap.exe END %1=%2 %3=%4 %5=%6 %7=%8
EXIT 0
:Failed
REM Signal that activation has failed
wzudpap.exe FAIL %1=%2 %3=%4 %5=%6 %7=%8 TEXT="Activation error reason"
EXIT %RC%
```

Linux/Unix Shell Scripts

You may use Linux/Unix shell scripts to perform deploy package activation processing and for Pre/Post exit processing. You may **not** use shell scripts for deploy implementation processing.

Shell scripts have no direct access to the ISPW server, and will need to report the status of the activation by adding entries to the deploy package log file.

Use the wzudpap command line utility to create the log, add entries to it, and report activation status. Refer to [WZUDPAP – Writing to the Log](#) on page 70 for details.

When using shell scripts for activation processing, you will need to make use of the Deploy Monitor feature of the Remote Server. Refer to [Deploy Activation Monitor](#) on page 68 for details.

Shell Script Example

[Figure 67](#) on page 68 shows a sample Linux/Unix shell script for package activation.

Figure 67 Linux/Unix shell script

```
#!/bin/sh
# Sample Deploy Activation Script
# Syntax: "sample.sh DREQID=nnn PKGID=nn SYSTNAME=xxx LOGPATH=lll SRID=sss"
# Signal that activation has started
./wzudpap START $1 $2 $3 $4 $5
# Do activation stuff here. Set "RC" to return code
RC=0
./wzudpap MSG $1 $2 $3 $4 TEXT="This message added to deploy log"
if [ $RC -eq 0 ]
then
# Signal that activation has completed
./wzudpap END $1 $2 $3 $4
exit 0
else
# Signal that activation has failed
./wzudpap FAIL $1 $2 $3 $4 TEXT="Activation error reason"
exit $RC
fi
```

Deploy Activation Monitor

The deploy process monitor is a feature of the Remote Server. It can be used to set options used for the deploy processes started by the Remote Server and monitor for completion of these processes. If the process is a bat/shell script, the deploy monitor can also send package completion status back to the ISPW server.

The deploy monitor makes use of a control file called dpapgm.ini. This file should be located in the directory specified by the "COMMAND_PATH" value in the Remote Server start up parameters.

One or more entries can be added to this file that define options for deploy processes started by the Remote Server. Each time the Remote Server will start a deploy process, it checks this file for an entry matching the process name to be started.

[Figure 68](#) shows what an entry in the dpapgm.ini may contain.

Figure 68 Sample dpapgm.ini Entry

```
[process-name]
monitor = yes/no
complete_on_start = yes/no
program = command-file-name
start_time_max = nnn
run_time_max = nnn
userid = user-id
```

[Table 34](#) describes the parameters.

Table 34 dpapgm.ini Parameters

Parameter	Description
[process-name]	the 8 character name that is defined in the deploy environment setup (i.e. M.DV).
monitor	specifies if the Remote Server should monitor the started process for completion, and pick up the deploy package status and forward it to ISPW server. If the process command type is a PHP script, then this value is ignored. The default is "no".
complete_on_start	specifies if the Remote Server should update the deploy package status to complete as soon as the process is successfully started. If "monitor" is set to "yes", then this value is ignored. The default is "no".

Table 34 dpapgm.ini Parameters (*Continued*)

Parameter	Description
program	used to over-ride the name of the program file that will be started. This can be used to run commands with names longer than 8 characters or command files that end in suffixes other than the ones that the Remote Server checks for. If this is not specified, then the process name is used to create command file name that will be started by the Remote Server.
start_time_max	specifies process start timeout in seconds. If "monitor" is set to "yes", then the Remote Server will monitor to see if the process has started. The process will be considered started when the deploy log gets created. If "monitor" is set to "no", then this value is ignored.
run_time_max	specifies process execution timeout in seconds. If "monitor" is set to "yes", then the Remote Server will monitor to see if the process has ended. The process will be considered ended when the deploy log contains the "Set_Status" message. The status from the deploy log will be forwarded to the ISPW server. If "monitor" is set to "no", then this value is ignored.
userid	specifies the local id to run the process under. This is only valid on Linux and Unix systems. If this is not specified, then the process will run under the same local id as the Remote Server.

DPAPGM.INI Example

[Figure 69](#) shows an example of the dpapgm.ini file contents. A copy of this file can be found in the samples subdirectory of the Remote Server install directory.

Figure 69 dpapgm.ini Example

```
[dpacmd1]
monitor = yes
program = dp_script1.sh
start_time_max = 5
run_time_max = 10
userid = *
;
[dpacmd2]
monitor = yes
program = dp_script2.sh
start_time_max = 5
run_time_max = 60
userid = userabc
;
[default]
monitor = no
complete_on_start = no
```

If no match is found for [process], then the values in the [default] entry are used.

Monitoring

A monitoring/reporting facility can also be used in conjunction with an activation log utility program to report activation results back to ISPW from within a bat file or shell script. This is not a replacement for specific activation programs, but an option for custom activation processing. When the monitoring option is enabled, the Remote Server will watch for the creation of the deploy activation log file. If the log is not created within the start_time_max, then CM is notified of the failure. Once the log has been created, then the Remote Server will read it periodically to see if the activation has been marked as complete or failed. When the results appear in the log, they are reported for CM. If no results are reported within the run_time_max, then CM is notified of the failure.

WZUDPAP – Writing to the Log

A command line utility called `wzudpap` is provided to create and write messages to the deploy package log file. The program is invoked with a set of parameters that tell it what to do. The parameters also tell it where to locate the log file. The format of the command line is:

```
wzudpap opt DREQID=nnn PKGID=nn SYSTNAME=target LOGPATH=logpath TEXT="text message"
```

“*opt*” specifies one of the following options:

- “START” to create the log file
- “MSG” to write a text message to the log file
- “END” to set the package status to complete
- “FAIL” to set the package status for failed.

```
“DREQID=nnn PKGID=nn SYSTNAME=target LOGPATH=logpath”
```

This set of values is used by the utility to find the location and name of the deploy package log file. These values are generally passed as parameters to the batch command or shell script that makes use of this utility.

“TEXT” specifies the text message to be inserted on a “MSG” option. It is also used to contain an error message on a “FAIL” option.

Example Usage for WZUDPAP

[Figure 70](#) shows an example of a bat file using the utility.

Figure 70 Example usage for WZUDPAP

```
REM Sample Deploy Activation Command
REM Syntax: "sample.bat DREQID=nnn PKGID=nn SYSTNAME=xxx LOGPATH=111 SRID=sss"
REM Signal that activation has started
wzudpap.exe START %1=%2 %3=%4 %5=%6 %7=%8 %9=%10
REM Do activation stuff here. Set "RC" to return code
wzudpap.exe MSG %1=%2 %3=%4 %5=%6 %7=%8 TEXT="This message added to deploy log"
IF %RC% GTR 0 GOTO :Failed
:Complete
REM Signal that activation has completed
wzudpap.exe END %1=%2 %3=%4 %5=%6 %7=%8
EXIT 0
:Failed
REM Signal that activation has failed
wzudpap.exe FAIL %1=%2 %3=%4 %5=%6 %7=%8 TEXT="Activation error reason"
EXIT %RC%
```

Each invocation of the WZUDPAP utility will append one or more lines of text to the activation log file. The monitor will pick these up and report status back to CM.

Example Log

[Figure 71](#) on page 71 shows what the activation log would look like in the above example.

Figure 71 Activation Log Example

```
Deploy Activation Started: Wed Aug 11 09:52:20 2013

DREQID:1997
PKGID: 1

This message added to deploy log
Set_Status=C

Deploy Activation Ended: Wed Aug 11 09:52:20 2013
```


Security

Overview

ISPW utilizes the security strengths of the z/OS platform to provide a secure environment for all processes in the development life-cycle right through to implementation into production.

Deployment across physical machines raises new security concerns that must be addressed. ISPW Deploy builds upon and utilizes the underlying security strengths of ISPW to ensure that the deployment functions are performed in as secure an environment as possible.

Security Enhancements

ISPW Deploy has incorporated security in the following areas:

- [Deploy Reference Data](#)
- [Component Transport Security](#)
- [User Functions](#).

Deploy Reference Data

ISPW Deploy has extended the ISPW Reference Data to include the following:

- Deployment Systems
- Deployment Domains
- Deployment Types
- Deployment Categories
- Deployment Environments.

The user functions that are used to define and maintain this reference data have been secured.

What has been added?

A new Security Object called DPLYREF has been created to protect this data separately to the base ISPW Reference data, which is protected by the Object REFDATA. [Table 35](#) lists the added methods.

Table 35 Security Methods

Method	Usage	Default Security Check	Available Variables
SYSTEM	Controls who can maintain Deployment Systems	<Server>.SYSTEM.<Systnam>.<Systtyp> Access: UPDATE	Systnam Systtyp
CATEGORY	Controls who can maintain Deployment Categories	<Server>.CATEGORY.<Dpcat> Access: UPDATE	Dpcat
DOMAIN	Controls who can maintain Deployment Domains	<Server>.DOMAIN.<Dpdmn> Access: UPDATE	Dpdmn
TYPE	Controls who can maintain Deployment Types	<Server>.TYPE.<Dptype>.<Dpcat> Access: UPDATE	Dptype Dpcat
ENV	Controls who can maintain Deployment Environments	<Server>.ENV.<Dpenv>.<Owner> Access: UPDATE	Dpenv Owner

See the *ISPW Technical Reference Guide* chapter entitled “Security” for a complete description of how this security feature works.

Component Transport Security

This relates to the security around the Component Transport (CT) tasks. These are MVS Started Tasks that are responsible for the warehousing of components and the movement of components and their parts between datasets and machines.

MVS STC

The CT Tasks are MVS Started Tasks and therefore inherit all of the security associated with having an isolated address space with its own UserID.

CT Messages

ISPW messages are sent to CT Tasks that contain the instructions that the CT Task is to perform. These messages are secured by the following means:

- CT Identification
- Security Tokens

CT Identification

Each CT Task has a parameter file that tells it which ISPW Master Address space (CM Task) it is to communicate with. The CT Task sends a message to the CM Task, and the CM Task will perform two security checks before it can know that this is in fact a CT Task:

- SERVER LOGON – that standard ISPW logon that all ISPW users must be authorized to; and
- SERVER CTIDENT – this identifies that the UserID of the CT Task is authorized and identified as a CT Task.

See the *ISPW Technical Reference, Chapter 7 – Security*, for a complete description of these Security Rules.

CT Security Token

Once the CM Task has identified the CT Task, it passes a security token to CT. All future messages with instructions for that CT Task must have the same security token; otherwise the CT Task will reject the message. The header of the message that contains the Security token (among other things) is encrypted.

User Functions

Deploy has introduced some new User functions that are required to be secured.

Protecting Deploy Requests

Deploy Requests are the “unit of Deployment” and all deployments are performed via one of these Requests. A new security Object has been created to protect Deploy Requests.

A new Security Object called DPLYREQ has been created to protect the Deploy Request information. [Table 36](#) lists the added Methods.

Table 36 Protection Methods

Method	Usage	Default Security Check	Available Variables
RESTART	Controls who can restart a Deployment Request	<Server>.DPLYREQ.<Dpenv> Access: UPDATE	Dpenv
CANCEL	Controls who can cancel a Deployment Request	<Server>.DPLYREQ.<Dpenv> Access: UPDATE	Dpenv
TERMINAT	Controls who can terminate a Deployment Request	<Server>.DPLYREQ.<Dpenv> Access: UPDATE	Dpenv
PKGFAIL	Controls who can fail a Package within a Deployment Request	<Server>.DPLYREQ.<Dpenv> Access: UPDATE	Dpenv

See the *ISPW Technical Reference Guide* chapter entitled “Security” for a complete description of how this security feature works.

User Reference

Overview

This chapter explains Deployment for the User and includes the following sections:

- [Deployment Requests](#)
- [Deploy Request Queue](#)
- [Task List Options and Display](#)
- [Deploy Set Processing.](#)

Deployment Requests

This section explains the life-cycle of the Deployment Request for the User of ISPW Deploy.

When a Promote, Generate or Implement operation is present in a Set, ISPW will check at Set Lock time if any Deployment Environments are specified for the target level of the operation.

If one or more Deployment Environments are specified then ISPW will create a Deployment Request per Environment if any of the component parts has an Implement Option of 'Y' at the target level of the operation and one or more Sub-environments are defined to which that component part is to be deployed.

Breaking a Request into Packages

Parts to be deployed are split into Packages within the Request. A Package will be created for each combination of Deployment Environment, Sub-environment, Category and Implementation Process for which any component parts need to be deployed.

Deployment Phases

A Deployment goes through various phases as summarized in [Table 37](#).

Table 37 Deployment Phases

#	Phase	Description
1.	Create Deployment Request	During the "Set Lock" process, the Deployment Request is created.
2.	Confirm Request	During the Execution for the Set, each deploy part is confirmed as being correct. Once all parts are confirmed then the Request is eligible for the next phase.
3.	Staging	If staging of the deployment has been defined, then immediately that the Request is Confirmed, the parts will be copied to the target warehouse.
4.	Implementation	Once the Implementation Date/Time is reached for each Package, the components are implemented according to the process as defined in the Deploy Reference data. If no Activation process has been defined, then after all Packages are successfully implemented the Request status will be set to "Complete".
5.	Activation	If an Activation process has been defined, then once the Activation Date/Time is reached for each Package, the components are activated according to the process as defined in the Deploy Reference data. Once all Packages are successfully activated the Request status will be set to "Complete".

Deploy Request Queue

The Deploy Request Queue is the main entry panel into the Deploy Request data. [Table 38](#) lists the ways the Deploy Request Queue can be accessed.

Table 38 Accessing the Deploy Request Queue Methods

#	Method
1.	DP off the Main Menu.
2.	DP against a task that is deploying. However this will show only the Deployment Requests for the Set that the task is in. If the Task is not in a Set, no Requests will be displayed.
3.	DP against a Set that has deploying tasks. However this will show only the Deployment Requests for the Set.

The Deploy Request Queue is the entry point to all the Deployment activity in the system, and it contains a range of functions that are described in this section of the chapter. [Table 39](#) lists the functions.

Table 39 Deploy Request Queue Functions

#	Function
1.	Deploy Request Main Panel
2.	Deploy Request Messages
3.	Deploy Request Modify
4.	Deploy Request Packages - Composite view
5.	Deploy Request Packages - Summary view
6.	Deploy Request Packages - System view
7.	Deploy Request Package – List Items
8.	Deploy Request Package – View Details
9.	Deploy Request Package – Fail Package

Deploy Request Main Panel

Main menu with DP option highlighted is shown in [Figure 72](#). Enter option DP.

Figure 72 Main Menu with DP Option

```

ISPW 18.02                ISPW INT  ALLOCATIONS
OPTION ==>> DP

  U  USER                  - User Preferences          USERID - CRAIG
  A  ANALYSIS              - Cross Reference Analysis  USERID - CRAIG
  P  CONTAINERS            - Container List             PREFIX  - CRAIG
  W  WORK                  - Work List                  TIME    - 03:02
  R  REPOSITORY            - Repository List            DATE    - 2016/07/29
  DP DEPLOY                - Deploy Requests            - 2016.211
  GI GENERATE IMPACTS     - Generate Impacts          PROC    - ISPFPROC
  M  MAINTENANCE          - Maintenance Functions     TABLES - W3T
  Z  FUNCTIONS            - Add Special Functions (Z1 - Z8)
                                SERVER  - W3T

Z1
Z2
Z3
Z4
Z5
Z6
Z7
Z8

Enter END command to return to the Primary Menu.

```

The Deploy Request screen is shown in [Figure 73](#). Make sure the View field is set to R.

Figure 73 Deploy Request Screen

ReqId	Set	DplyEnv	Status	Description	Date	MM	DD	Time
00450	S000002594	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:44		
00449	S000002594	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:44		
00448	S000002594	DEMOPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:44		
00447	S000002594	DBSPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:44		
00445	S000002593	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:40		
00444	S000002593	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:40		
00443	S000002593	DEMOPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:40		
00442	S000002593	DBSPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:40		
00440	S000002592	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:36		
00439	S000002592	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:36		
+ 00438	S000002592	DEMOPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:36		
00437	S000002592	DBSPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14:36		
+ 00436	S000002591	PAUL	Terminate	PAUL PAYROL CHANGE	2016-09-24	14:32		
00431	S000002590	PAUL	Completed	PAUL PAYROL CHANGE	2016-09-24	10:10		
00426	S000002589	ACTST	Completed	GL LEDGER01	2016-09-23	14:59		
00425	S000002582	ACTST	Completed	GL LEDGER01	2016-09-23	14:14		
00423	S000002580	ACTST	Completed	GL LEDGER01	2016-09-23	14:12		
00422	S000002579	ACTST	Completed	GL LEDGER01	2016-09-23	14:11		
00421	S000002575	ACTST	Planned	GL LEDGER01	2016-09-23	11:12		
00420	S000002575	SYSAT	Planned	GL LEDGER01	2016-09-23	11:12		
+ 00417	S000002572	ACTST	Completed	GL LEDGER01	2016-09-23	08:53		
+ 00416	S000002571	ACTST	Completed	GL LEDGER01	2016-09-23	07:56		
00415	S000002570	ACTST	Completed	GL LEDGER01	2016-09-23	07:40		
+ 00414	S000002569	DON2	Failed	API TESTING	2016-09-22	16:35		

View

The View field as highlighted in [Figure 73](#) allows you to switch deploy views. [Table 40](#) lists the available values.

Table 40 View Input Values

Value	Description
R	Deploy Request View
S	Deploy System View

Filter Dates

The Filter Dates line as highlighted in [Figure 73](#) is used to control the Request Queue display. [Table 41](#) lists the available values for the input control field.

Table 41 Filter Dates Input Values

Value	Description
T	Today – Requests created today
D	Day – Requests created in the last day
W	Week – Requests created in the last week
M	Month – Requests created in the last month
R	Date Range – Requests created between the range of dates specified on the Filter Line.
N	No Range – All Requests will be listed.

On entering the request queue for the first time, the filtering will be set to W. If the User preferences (option U.S) are set to “Session Save = Y” then whatever value is typed in the filter line will be saved across sessions.

Columns

[Table 42](#) lists the columns on the Request Screen.

Table 42 Request Screen Columns

Column	Description
ReqId	Unique ID that identifies a Request
Set	ISPW Set that the Deploy Request was created for
DplyEnv	Deploy Environment
Status	Overall Status of the Request. Table 43 lists possible Status values
Description	Taken from the Set Description
Date/Time	Date/Time that the Request was created

Note that by using the VP (View Preferences) command, the column positions and display lengths can be altered to the User’s preference.

Deployment Request Status

[Table 43](#) lists the different request statuses:

Table 43 Deployment Request Status

Status	Description
Planned	After the “Set Lock” process, the Deployment Request is created, and has a status of Planned to indicate that this request contains the intended list of parts for deployment.
Confirmed	Once the Set has executed, the Tasks are checked for completeness and the request status is updated to Confirmed, which means that they are ready for the Deployment process.
Staged	If a staging warehouse is defined, then staging will be done, where ISPW Deploy will transfer the component parts to the target platform and store them in a warehouse ready to be deployed.
Implemented	For deployments with an Activation process, a status of Implemented indicates that the Implemented Process is complete.
Completed	This indicates that the whole deployment process for this request is complete, whether it had Activation or not.
Failed	A Package (or multiple packages) has a Failed status.
Terminated	The Deployment Request has been manually terminated.

Operations Against a Request

[Table 44](#) lists the valid operations.

Table 44 Request Screen Line Commands

Op	Description
S	Select Request to show Package detail. See Deploy Request Packages - Composite View .
L	List Messages. See Deploy Request Messages .
M	Modify Request. See Deploy Request Modify .

Sections later in this chapter describe each of these operations.

Deploy Request Messages

Messages are stored against the Deployment Request when an error has occurred and when a manual User action has been executed. Requests with messages are identified by the “+” next to the RequestID.

Figure 74 Deploy Request Queue Screen

```

ISPW 18.02                                DEPLOY REQUEST QUEUE                                Row 1 of 262
Command ==>                                Scroll ==> CSR

View: R Filter Dates: W (From: 2016-08-25 To: 2016-09-25 )

```

ReqId	Set	DplyEnv	Status	Description	Date	MM	DD	Time
00450	S000002594	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	44	
00449	S000002594	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	44	
00448	S000002594	DEMOPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	44	
00447	S000002594	DBSPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	44	
00445	S000002593	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	40	
00444	S000002593	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	40	
00443	S000002593	DEMOPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	40	
00442	S000002593	DBSPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	40	
00440	S000002592	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	36	
00439	S000002592	ISPFPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	36	
+ 00438	S000002592	DEMOPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	36	
00437	S000002592	DBSPROC	Completed	PAUL PAYROL CHANGE	2016-09-24	14	36	
L + 00436	S000002591	PAUL	Terminate	PAUL PAYROL CHANGE	2016-09-24	14	32	
00431	S000002590	PAUL	Completed	PAUL PAYROL CHANGE	2016-09-24	10	10	
00426	S000002589	ACTST	Completed	GL LEDGER01	2016-09-23	14	59	
00425	S000002582	ACTST	Completed	GL LEDGER01	2016-09-23	14	14	
00423	S000002580	ACTST	Completed	GL LEDGER01	2016-09-23	14	12	
00422	S000002579	ACTST	Completed	GL LEDGER01	2016-09-23	14	11	
00421	S000002575	ACTST	Planned	GL LEDGER01	2016-09-23	11	12	
00420	S000002575	SYSAT	Planned	GL LEDGER01	2016-09-23	11	12	
+ 00417	S000002572	ACTST	Completed	GL LEDGER01	2016-09-23	08	53	
+ 00416	S000002571	ACTST	Completed	GL LEDGER01	2016-09-23	07	56	
00415	S000002570	ACTST	Completed	GL LEDGER01	2016-09-23	07	40	
+ 00414	S000002569	DON2	Failed	API TESTING	2016-09-22	16	35	

Five Requests in the Request Queue shown in [Figure 74](#) on page 81 are marked with a “+”, indicating that they have messages. Enter L against the Request or place the cursor on the “+” and press Enter to display the messages for a request.

Message Screen

[Figure 75](#) shows the messages for a selected request.

Figure 75 Deploy Request Messages

```

ISPW 18.02                DEPLOY REQUEST: MESSAGES                Row 11 of 25
Command ==>                Scroll ==> CSR

+-----+
| Review Exception and Warning Conditions reported by ISPW Deploy. |
| Set: S000002591 - PAUL DEPLOY TESTING                            |
| Press ENTER or END to continue.                                  |
+-----+

                          Exception and Warning Conditions.
-----
E: Dataset not found: DSN=SYS4.DEPLOY.MKT.JOBS
E: Deploy processing failed for package: Req=436/3/MP3000. One of the
  items in the package has failed. Check the Component Transport log
  for details
I: Tue Sep 24 14:35:12 2002: RESTART requested by user PAUL
E: Dataset not found: DSN=SYS4.DEPLOY.MKT.JOBS
E: Deploy processing failed for package: Req=436/3/MP3000. One of the
  items in the package has failed. Check the Component Transport log
  for details
I: Tue Sep 24 14:35:19 2002: RESTART requested by user PAUL
E: Dataset not found: DSN=SYS4.DEPLOY.MKT.JOBS
E: Deploy processing failed for package: Req=436/3/MP3000. One of the
  items in the package has failed. Check the Component Transport log
  for details
I: Fri Jan 24 14:41:12 1975: CANCEL requested by user PAUL
-----
                          Bottom of List -----

```

Errors are displayed in red and Informational messages are displayed in green. The informational messages in the above example show the attempted Restarts and the Cancel that eventually terminated the request.

Deploy Request Modify

The Request Modify function is invoked by entering an **M** against the request.

Figure 76 Deploy Request Modify

```

ISPW                      MODIFY DEPLOY REQUEST 414
COMMAND ==>

Description ==> Payroll Updates
Owner       ==> DONEGEN                      Release ==>
Change Type ==> S                          Create  ==> 2002-09-22 16:35:03
Operation   ==> P                          Start   ==> 2002-09-22 16:35:00
Level      ==> DSI                         Implement ==> 2002-09-22 16:35:00
Set        ==> S000002569                  Activate ==> 2002-09-22 16:35:00
Application ==> DON1                        Queue   ==>
Stream     ==> DON1                        Job/Proc ==> WZDW3 / WZDW3
Environment ==> DON2                        Class   ==> D

Set Status      ==> Deploy Failed
Deploy Request Status ==> Failed

-----

Deploy Control ==> (RESTART TERM)

Press Enter to continue, or End to terminate.

```

[Figure 76](#) displays information about the Deploy Request and the ISPW Set that it was created from.

Deploy Control

Depending upon the Deploy Request Status, one of the commands listed in [Table 45](#) may be entered.

Table 45 Deploy Control Commands

Command	Description	Statuses
RESTART	Restarts the deployment after a failure.	Failed
CANCEL	Stop a Request from processing and Terminate it.	Confirmed Staged Implemented
TERMINATE	Terminate a Request after a failure.	Failed (however no Packages can be executing)
No Command possible	Deploy Control will not allow any command.	Planned Completed Terminated

For situations where no command is possible it may be necessary to Fail a Physical Package, which will then cause the overall Request status to be “Failed”.

Deploy Request Packages - Composite View

Select a Request

Selecting a Deploy Request displays the Deploy Request Packages screen, which has three views. The first view is the “Composite” view ([Figure 77](#) on page 83). The view can be changed by placing the cursor on the view name and pressing Enter.

Figure 77 Deploy Request Packages - Composite view

```

ISPW 18.02          DEPLOY REQUEST - 0000000053          Row    1 of 12
Command ==>                               Scroll ==> CSR
                                           More -->
+-----+-----+-----+-----+-----+-----+-----+-----+
| View    ==> Composite Summary System |
| Description: LONG BEACH - IVP         |
| Environment: PRODEXEC Set: S000001590 |
+-----+-----+-----+-----+-----+-----+-----+
| PkgId Sub System  DplyCat  Status      DateStrt   TimeStrt  DateEnd  |
+-----+-----+-----+-----+-----+-----+-----+
| 00001 SYSA          BATCH   Completed   2016-02-21 11:24:21 2016-02-21 |
|          CALGARY    IM:Completed 2016-02-21 11:24:23 2016-02-21 |
|          GABRIOLA   IM:Completed 2016-02-21 11:24:23 2016-02-21 |
|          LONDON     IM:Completed 2016-02-21 11:24:23 2016-02-21 |
| 00002 SYSB          BATCH   Completed   2016-02-21 11:24:21 2016-02-21 |
|          CALGARY    IM:Completed 2016-02-21 11:24:23 2016-02-21 |
|          GABRIOLA   IM:Completed 2016-02-21 11:24:23 2016-02-21 |
|          LONDON     IM:Completed 2016-02-21 11:24:23 2016-02-21 |
| 00003 SYSC          BATCH   Completed   2016-02-21 11:24:21 2016-02-21 |
|          CALGARY    IM:Completed 2016-02-21 11:24:23 2016-02-21 |
|          GABRIOLA   IM:Completed 2016-02-21 11:24:23 2016-02-21 |
|          LONDON     IM:Completed 2016-02-21 11:24:23 2016-02-21 |
+-----+-----+-----+-----+-----+-----+-----+
Bottom of List
    
```

This screen can be scrolled to the right. Note that the column order may be different between users. The column order can be changed via the VP command.

Logical and Physical Packages

This screen shows both the Logical and Physical packages (which is why it is called Composite). The Logical Package is identified on the line with the PackageID. Each line below that containing a different System represents a separate Physical Package.



A Logical Package represents the deployment of a set of component parts to one or more Targets. The Physical Package represents the deployment of those components to one Target System only. Most sites will probably deploy to a single Target and so this concept can easily be missed.

Columns

The columns and their descriptions on the Request Package screen are listed in [Table 46](#).

Table 46 Request Packages Columns

Column	Description
PkgId	Unique ID that identifies a Package
Sub	Deployment Sub-environment
System	Target System
Date Strt Time Strt	Date and Time that the Package began deployment.
Date End Time End	Date and Time that the Package successfully ended deployment.
DplyCat	Deploy Category
Status	The Logical Package status is derived from the individual Physical Package statuses. These statuses are described later in this section.
Date/Time Impl	Date/Time that the Deployment Implementation is to begin.
Date/Time Actv	Date/Time that the Deployment Activation (if any is defined) is to begin

Logical Package Statuses

This status is derived from the statuses of all the Physical packages. [Table 47](#) lists possible status values'

Table 47 Request Packages Status

Status	Description
<i>blank</i>	No processes have yet been identified.
Staged	All deployable parts are staged onto their respective Target systems.
Implemented	This Package undergoes Activation, and the Implementation processes are complete.
Completed	All Deploy processes are complete (i.e. if both Implementation and Activation are defined then both are completed successfully. If only Implementation is required then this has completed successfully).
Failed	One or more Physical Packages has failed.

Physical Package Statuses

The Physical Package Status is constructed of two parts:

XX:Status

Where XX is a two character representation of the process that the status relates to. [Table 48](#) lists Status Process and their statuses are as follows:

Table 48 Physical Package Status

Process	Status	Description
ST Staging	Wait Time	This status is not commonly seen for Staging as there is no specific Date/Time constraint for the Staging process. A Physical Package can have this status for 30 seconds or less due to the internal Deploy Scheduler
	Ready	All conditions have been met for the Staging Process to begin. This status could indicate that the Target CT Task is not running.
	Dispatched	CT Task has been woken up to start Staging
	Executing	CT Task is performing the Staging process
	Complete	Staging is complete
	Failed	Staging has failed
	Terminated	Indicates that the overall Request was terminated for Packages that were not Complete or Failed when the Terminate was issued.
IM Implementing and AC Activating	Wait Time	Implementation is waiting on the Implementation Time or Activation is waiting on Activate Time.
	Ready	All conditions have been met for the Implement or Activate process to begin. This status could indicate that the Target CT Task is not running.
	Dispatched	CT Task has been woken up to start the process
	Held	Process has been started in a HELD status – option not currently supported.
	Wait External	An External process has been defined and ISPW is waiting for it to start.
	Executing	Implementation/Activation is executing
	Complete	Implementation/Activation process is complete
	Failed	Implementation/Activation process has failed.
	Terminated	Indicates that the overall Request was terminated for Packages that were not Complete or Failed when the Terminate was issued.

Operations Against a Package

[Table 46](#) lists the valid operations against a Logical Package.

Table 49 Operations Against a Package

Op	Description
S	Select Package to list Items
M	Select Package details for viewing

Deploy Request Packages - Summary view

Select a Deploy Request and then select the view “Summary” to display the screen shown in [Figure 78](#).

Figure 78 Deployment Request - Summary View

```

ISPW 18.02          DEPLOYMENT REQUEST - 0000000053          Row    1 of    3
Command ==>                               Scroll ==> CSR
                                           More -->
+-----+-----+-----+-----+-----+-----+-----+-----+
| View   ==> Composite Summary System
| Description: LONG BEACH - IVP
| Environment: PRODEXEC Set: S000001590
+-----+-----+-----+-----+-----+-----+-----+-----+
PkgId Sub  System  DplyCat  Status      DateStrt    TimeStrt    DateEnd
-----+-----+-----+-----+-----+-----+-----+-----+
00001 SYSA          BATCH   Completed   2016-02-21  11:24:21    2016-02-21
00002 SYSB          BATCH   Completed   2016-02-21  11:24:21    2016-02-21
00003 SYSC          BATCH   Completed   2016-02-21  11:24:21    2016-02-21
+-----+-----+-----+-----+-----+-----+-----+-----+
Bottom of List

```

Logical Packages only

This view shows only the Logical Packages. The status is an overall status derived from the status of each Physical deployment taking place for each Package.

Columns and Status values

See [Table 46](#) on page 84 through [Table 48](#) on page 85 for complete description of the columns and statuses.

Operations against a Package

The following operations are valid against a Logical Package:

Table 50 Operations Against a Package

Op	Description
S	Select Package to list Items
M	Select Package details for viewing

Deploy Request Packages - System View

Select a Deploy Request and then select the view “System” to display the System View screen ([Figure 79](#)).

Figure 79 Deploy Request System View

```

ISPW 18.02          DEPLOY REQUEST - 0000000053          Row    1 of    9
Command ==>          Scroll ==> CSR
                               More -->
-----+-----
| View    ==> Composite Summary System
| Description: LONG BEACH - IVP
| Environment: PRODEXEC Set: S000001590
|-----+-----
PkgId Sub  System  DplyCat  Status      DateStrt  TimeStrt  DateEnd
-----+-----+-----+-----+-----+-----+-----+-----
00001 SYSA  CALGARY  BATCH    IM:Completed 2016-02-21 11:24:23 2016-02-21
00001 SYSA  GABRIOLA BATCH    IM:Completed 2016-02-21 11:24:23 2016-02-21
00001 SYSA  LONDON   BATCH    IM:Completed 2016-02-21 11:24:23 2016-02-21
00002 SYSB  CALGARY  BATCH    IM:Completed 2016-02-21 11:24:23 2016-02-21
00002 SYSB  GABRIOLA BATCH    IM:Completed 2016-02-21 11:24:23 2016-02-21
00002 SYSB  LONDON   BATCH    IM:Completed 2016-02-21 11:24:23 2016-02-21
00003 SYSC  CALGARY  BATCH    IM:Completed 2016-02-21 11:24:23 2016-02-21
00003 SYSC  GABRIOLA BATCH    IM:Completed 2016-02-21 11:24:23 2016-02-21
00003 SYSC  LONDON   BATCH    IM:Completed 2016-02-21 11:24:23 2016-02-21
-----+-----+-----+-----+-----+-----+-----+-----
Bottom of List -----
    
```

Physical Packages Only

This view shows all of the Physical Packages. In the example in [Figure 79](#), each Package is deployed to two Target systems.

Columns and Status Values

See [Table 46](#) on page 84 through [Table 48](#) on page 85 for complete description of the columns and statuses.

Operations Against a Package

[Table 51](#) lists the valid operations against a Physical Package.

Table 51 Operations Against a Physical Package

Op	Description
S	Select Package to list Items
M	Select Package details for viewing and manual "FAIL"

Deploy Request Package – List Items

Select a Logical Package

Selecting a Logical Package from the Composite view displays the following screen ([Figure 80](#)). This shows both the Logical and Physical Item Lists in a composite view.

Figure 80 Deploy Request Logical and Physical Item List

```

ISPW 18.02          DEPLOYMENT REQUEST - 0000000049          Row 1 of 15
Command ==>                               Scroll ==> CSR
                                           More -->
+-----+-----+-----+-----+-----+-----+-----+-----+
| View   ==> Composite Summary System
| Description: DEPLOY TESTING
| Environment: ACTST   Implementation Date/Time: Not Specified
| SubEnv  : CT       Activation   Date/Time: Not Specified
| Category : BATCH   Status      : Completed
+-----+-----+-----+-----+-----+-----+-----+

  ItmId PrtId Name      DplyType Appl Type Clas A System  Status
-----+-----+-----+-----+-----+-----+-----+-----+
  00001 00552 TPROG01  LOADBAT  ABN  LOAD  BAT          ABN-FA  Completed
                                           ABN-FB  Completed
  00002 00553 TPROG02  LOADBAT  ABN  LOAD  BAT          ABN-FA  Completed
                                           ABN-FB  Completed
  00003 00554 TPROG03  LOADBAT  ABN  LOAD  BAT          ABN-FA  Completed
                                           ABN-FB  Completed
-----+-----+-----+-----+-----+-----+-----+
                                           Bottom of List

```

Scroll this screen to the right to view the remaining columns.

Columns

[Table 52](#) lists the columns on the Item List screen.

Table 52 Item List Columns

Column	Description
ItemId	Unique ID that identifies a Part within a Package
Part	The Part Name being deployed
DplyType	Deploy Category
Appl	ISPW Application
Type	Part Type
Class	Part Class
A	ISPW Action (For example: D for Delete, F for Fallback)
System	Target System
Status	There is a "Logical Item Status" which is the status on the same line as the ItemID. This is an overall status derived from the individual "Physical Item Status" that is listed against each Target System.
Storage Name	This is the Dataset Name/HFS Path that the Part is to be deployed to.

Logical Item Status

This status is derived from the statuses of all the Physical items. [Table 53](#) lists the possible status values.

Table 53 Logical Item Status

Status	Description
<i>blank</i>	No processes have yet been identified.
Staged	Staging has been completed for all Target systems for this part.
Implemented	This Item undergoes Activation, and the Implementation processes are complete.

Table 53 Logical Item Status (*Continued*)

Status	Description
Completed	All Deploy processes are complete for this Item (i.e. if both Implementation and Activation are defined then both are completed successfully. If only Implementation is required then this has completed successfully).
Failed	One or more Physical Item processes has failed.

Physical Item Status

Each Physical Item has a status as listed in [Table 54](#).

Table 54 Physical Item Status

Status	Description
<i>blank</i>	No processes have yet been identified.
Completed	All processes are complete for this physical Item (i.e. if both Implementation and Activation are defined then both are completed successfully. If only Implementation is required then this has completed successfully).
Failed	Staging, Implementation or Activation has failed for this physical item.

Summary View

Display the Summary View by placing the cursor on “Summary” and pressing Enter. The Logical Item List is displayed as shown in [Figure 81](#).

Figure 81 Deploy Request Logical Package Summary View

```

ISPW 18.02          DEPLOYMENT REQUEST - 0000000049          Row 1 of 5
Command ==>                               Scroll ==> CSR
                                           More -->
+-----+-----+-----+-----+-----+-----+-----+-----+
| View   ==> Composite Summary System
| Description: DEPLOY TESTING
| Environment: ACTST   Implementation Date/Time: Not Specified
| SubEnv  : CT        Activation      Date/Time: Not Specified
| Category : BATCH    Status          : Completed
+-----+-----+-----+-----+-----+-----+-----+

  ItmId PrtId Name      DplyType Appl Type Clas A System  Status
-----+-----+-----+-----+-----+-----+-----+
  00001 00552 TPROG01  LOADBAT  ABN  LOAD  BAT      Completed
  00002 00553 TPROG02  LOADBAT  ABN  LOAD  BAT      Completed
  00003 00554 TPROG03  LOADBAT  ABN  LOAD  BAT      Completed
-----+-----+-----+-----+-----+-----+-----+
                                           Bottom of List
    
```

[Figure 81](#) shows the list of items from a logical perspective. No operations are valid.

System View

Display the System View by placing the cursor on “System” and pressing Enter. The Physical Item List is displayed as shown in [Figure 82](#) on page 90.

Figure 82 Deploy Request Logical Package System View

```

ISPW 18.02          DEPLOY REQUEST - 0000000049          Row 1 of 10
Command ==>          Scroll ==> CSR
                               More -->
+-----+
| View ==> Composite Summary System
| Description: DEPLOY TESTING
| Environment: ACTST   Implementation Date/Time: Not Specified
| SubEnv : CT         Activation Date/Time: Not Specified
| Category : BATCH    Status : Completed
+-----+

  ItmId PrtId Name      DplyType Appl Type Clas A System  Status
-----
00001 00552 TPROG01  LOADBAT  ABN  LOAD  BAT  ABN-FA  Completed
00001 00552 TPROG01  LOADBAT  ABN  LOAD  BAT  ABN-FB  Completed
00002 00553 TPROG02  LOADBAT  ABN  LOAD  BAT  ABN-FA  Completed
00002 00553 TPROG02  LOADBAT  ABN  LOAD  BAT  ABN-FB  Completed
00003 00554 TPROG03  LOADBAT  ABN  LOAD  BAT  ABN-FA  Completed
00003 00554 TPROG03  LOADBAT  ABN  LOAD  BAT  ABN-FB  Completed
-----
Bottom of List -----
    
```

[Figure 82](#) shows the list of items from a physical perspective. Each row on the screen represents the physical deployment of a part.

Operations Against a Physical Item

[Table 55](#) lists the valid operations against a Physical Item.

Table 55 Operations Against a Physical Item

Op	Description
B	Browse the Part at the target location. ISPW will retrieve the part from the target dataset and display it in a temporary file in the User's session.

Deploy Request Package – View Details

Modify a Logical Package

Selecting (M) a Logical Package from the Composite or Summary views displays the [Modify Deploy Package](#) screen ([Figure 83](#)).

Figure 83 Modify Deploy Package

```

ISPW                                MODIFY DEPLOY PACKAGE 1
COMMAND ==>

Description      ==> TEST DEPLOY
Application      ==> PAYR
Set              ==> S000002598
Environment      ==> ACTST
Deploy Category  ==>
Implementation Process:
  Type           ==> C      Name ==>
  Job            ==>      Start ==> A      Date/Time ==> 2016-09-25 06:12:00
Activation Process:
  Type           ==>      Name ==>
  Job            ==>      Start ==>      Date/Time ==> 2016-09-25 06:12:00
Sequencing:
  Group          ==>      Seq. Number ==> 0   Seq. Scope ==>
Stream           ==> PAYR
Release          ==>
Sub Environment  ==> CT

Set Status              ==> Complete
Logical Package Status ==> Complete

-----

Press Enter to continue, or End to terminate.

```

No operations are possible from this screen.

Information Displayed

The information displayed includes:

- Set Details
- Deploy Request Details
- Implementation Process information
- Activation Process information.

Deploy Request Package – Fail Physical Package

Physical Packages can be manually “Failed” if necessary. Failing a Package should not be performed without good reason, and is only necessary when a process has failed such that ISPW Deploy cannot know of that failure.

Modify Physical Package

Entering **M** against a Physical Package takes you to the [Modify Deploy Package - Deploy Control FAIL](#) screen ([Figure 85](#) on page 92). This can only be entered from the “System” View for Deploy Packages ([Figure 84](#)).

Figure 84 Modify Physical Package

```

ISPW 18.02          DEPLOYMENT REQUEST - 0000000454          Row    1 of    4
Command ==>                               Scroll ==> CSR
                                           More -->
+-----+
| View    ==> Composite Summary System
| Description: TEST DEPLOY
| Environment: ACTST   Set: S000002598
+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| PkgId Sub  System  DplyCat  Status      DateImpl  TimeImpl  DateActv
+-----+-----+-----+-----+-----+-----+-----+-----+
| 00001 CT   SYS-FA  BATCH    IM:Completed 2016-09-25 06:12:00 2016-09-25
| 00001 CT   SYS-FB  BATCH    IM:Completed 2016-09-25 06:12:00 2016-09-25
| M 00002 II   SYS-FA  BATCH    IM:Wait Time 2016-09-25 06:12:00 2016-09-25
| 00002 II   SYS-FB  BATCH    IM:Completed 2016-09-25 06:12:00 2016-09-25
+-----+-----+-----+-----+-----+-----+-----+-----+
| Bottom of List
+-----+

```

This will present the following screen where “FAIL” can be entered in the Deploy Control input.

Figure 85 Modify Deploy Package - Deploy Control FAIL

```

ISPW              MODIFY DEPLOY PACKAGE 2
COMMAND ==>

Description      ==> TEST DEPLOY
Application      ==> SYS
Set              ==> S000002598
Environment      ==> ACTST
Deploy Category  ==> BATCH
Implementation Process:
  Type          ==> C      Name ==>
  Job           ==>      Start ==> A      Date/Time ==> 2016-09-25 06:12:00
Activation Process:
  Type          ==>      Name ==>
  Job           ==>      Start ==>      Date/Time ==> 2016-09-25 06:12:00

Set Status      ==> Deploy Staged
System SYS-FA  Status ==> IM:Wait Time

-----

Deploy Control   ==> FAIL      (FAIL)

Press Enter to continue, or End to terminate.

```

Entering “FAIL” in the Deploy Control input field will present the following screen:

Figure 86 Modify Deploy Package - Confirm Deploy Control FAIL

```

ISPW          MODIFY DEPLOY PACKAGE 2
COMMAND ==>

Description   ==> TEST DEPLOY
Application   ==> SYS
Set           ==> S000002598
Environment   ==> ACTST
Deploy Category ==> BATCH
Implementation Process:
  Type        ==> C      Name ==>
  Job         ==>      Start ==> A      Date/Time ==> 2016-09-25 06:12:00
Activation Process:
  Type        ==>      Name ==>
  Job         ==>      Start ==>      Date/Time ==> 2016-09-25 06:12:00

Set Status    ==> Deploy Staged
System SYS-FA Status ==> IM:Wait Time

-----

Confirm Package Fail ==> Y ( Y/N )
Press Enter to continue, or End to terminate.

-----+-----+
| WARNING: Deploy Package 2 will be Force Failed. You should only Force Fail a |
| package when you are sure that no deployment processes are active. Enter Y e |
| to confirm this request. |
+-----+-----+

```

You must confirm that the Package is to be Failed by entering **Y** in the Confirm Package Fail input.



Failing a Physical Package is only to be done if you understand the consequences, and this action is especially secured via the ISPW Security Rules. Failing a single Package does not undo any deploy processes for that Deploy Request that may have completed or be still in process.

Task List Options and Display

As an ISPW user, you generally work within your Tasklists. This section describes the ISPW Deploy features available within. Topics within this section include:

- [Task List Status Area](#)
- [Task OP – DP](#)
- [Task OP – DT](#)
- [Operation History – Deploy Operations.](#)

Task List Status Area

The Task List status area (highlighted in [Figure 87](#)) is used to provide information about Deploy.

Figure 87 Deploy Task List

```

ASSIGNMENT          SYS0000001: TEST DEPLOY          Row 1 of 2
Command ==>          Scroll ==> CSR
                    More -->
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Select( ) Add Approve Close Join Reset Show/Hide Work ++/-- |
+-----+-----+-----+-----+-----+-----+-----+-----+
  Type Name      Lev  Op A User      Appl Date MM  Time  Status
+-----+-----+-----+-----+-----+-----+-----+-----+
JOB  ADAPT01  ST   I   CRAIG   SYS  02-09-18  16:00  DP:    Staged
JOB  ADAPT02  ST   I   CLIVE   SYS  02-09-19  18:40  DP:    Failed
JOB  ADAPT03  ST   I   CLIVE   SYS  02-09-19  18:55  DP:    Staged
JOB  ADAPT04  CONS P   CLIVE   SYS  02-09-19  18:44
JOB  ADAPT05  ST   I   CRAIG   SYS  02-09-23  14:14
JOB  JCL001   ST   I   CLIVE   SYS  02-09-19  18:34  DP:    Failed
CLST REXX01   ST   I   CRAIG   SYS  02-09-04  08:55  SP:    S000002449
CLST REXX02   ST   I   CRAIG   SYS  02-09-25  06:37
CLST REXX03   ST   I   KARYNS  SYS  02-09-25  08:52
CLST REXX04   ST   I   CRAIG   SYS  02-09-25  06:37
CLST REXX05   ST   I   CRAIG   SYS  02-09-25  06:37  SP: I   Held
CLST REXX06   ST   I   CRAIG   SYS  02-09-25  06:37
CLST REXX07   ST   I   CRAIG   SYS  02-09-25  15:47  DP:    Staged
CLST REXX08   ST   I   CRAIG   SYS  02-09-25  15:55  DP:    Implement
CLST REXX09   ST   I   CRAIG   SYS  02-09-25  07:27

```

The example in [Figure 87](#) shows how Deploy information is shown in a similar way that Set Processing progress is displayed, but with the DP: prefix.

DP: Statuses

The status shown for Deploy is the “Set Deploy Status”, a new Status that is derived from the different statuses of all participating Deploy Requests (one per Deploy Environment). [Table 56](#) lists the possible Status values.

Table 56 DP Status

Status	Description
Confirmed	All Deploy Requests have been Confirmed
Staged	Staging is complete for all Deploy Requests
Implemented	Implementing is complete for all Deploy Requests – Activation is still required on at least one.
Completed	All Deploy functions are complete.
Failed	One or more Deploy Requests has Failed.

Terminated Requests

If a Deploy Request is Terminated, as far as the Set is concerned, deployment for that Request is complete.

Task OP – DP

Enter DP against a Task that is participating in Deployment.

Figure 88 Deploy Task List - DP

```

ASSIGNMENT          PAYR000001: TEST DEPLOY          Row    1 of   23
Command ==>          Scroll ==> CSR
                      More -->
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Select(/) Add Approve Close Join Reset Show/Hide Work +/- |
+-----+-----+-----+-----+-----+-----+-----+-----+
  Type Name      Lev  Op A User      Appl Date MM  Time  Status
-----+-----+-----+-----+-----+-----+-----+-----+
JOB  ADAPT01  ST   I   CRAIG   SYS  02-09-18  16:00  DP:   Staged
JOB  ADAPT02  ST   I   CLIVE   SYS  02-09-19  18:40  DP:   Failed
JOB  ADAPT03  ST   I   CLIVE   SYS  02-09-19  18:55  DP:   Staged
JOB  ADAPT04  CONS P   CLIVE   SYS  02-09-19  18:44
JOB  ADAPT05  ST   I   CRAIG   SYS  02-09-23  14:14
DP  JOB  JCL001  ST   I   CLIVE   SYS  02-09-19  18:34  DP:   Failed
      CLST REXX01  ST   I   CRAIG   SYS  02-09-04  08:55  SP:   S000002449
      CLST REXX02  ST   I   CRAIG   SYS  02-09-25  06:37
      CLST REXX03  ST   I   KARYNS  SYS  02-09-25  08:52
      CLST REXX04  ST   I   CRAIG   SYS  02-09-25  06:37
      CLST REXX05  ST   I   CRAIG   SYS  02-09-25  06:37  SP: I   Held
      CLST REXX06  ST   I   CRAIG   SYS  02-09-25  06:37
      CLST REXX07  ST   I   CRAIG   SYS  02-09-25  15:47  DP:   Staged
      CLST REXX08  ST   I   CRAIG   SYS  02-09-25  15:55  DP:   Implement
      CLST REXX09  ST   I   CRAIG   SYS  02-09-25  07:27
    
```

DP Screen

Entering the DP operation against a Task that is participating in Deployment displays the screen shown in [Figure 89](#).

Figure 89 DP Screen

```

ISPW 18.02          DEPLOY REQUEST QUEUE          Row    1 of    1
Command ==>          Scroll ==> CSR

  Filter Dates: R (From: 2001-08-25 To: 2002-09-25 )

  ReqId Set      DplyEnv  Status  Description          Date MM DD Time
-----+-----+-----+-----+-----+-----+-----+
+ 00379 S000002558 ACTST   Failed  TEST DEPLOY          2002-09-19 18:34
-----+-----+-----+-----+-----+-----+-----+
Bottom of List
    
```

[Figure 89](#) shows all Deploy Requests participating with the Set that the Task is in, and it is in Deploy Request Queue format.

Further Description

See [Deploy Request Queue](#) on page 78 for a description of all functions of the Deploy Request Queue.

Task OP – DT

Enter DT against a Task that has participated in Deployment.

Figure 90 Deploy Task List - DT

```

ASSIGNMENT          PAYR000001: TEST DEPLOY          Row 1 of 23
Command ==>          Scroll ==> CSR
                                More -->

Select(/) Add Approve Close Join Reset Show/Hide Work +/-

  Type Name   Lev  Op A User      Appl Date MM  Time  Status
-----
DT JOB ADAPT01 ST   I   CRAIG   SYS  02-09-18 16:00 DP:   Staged
   JOB ADAPT02 ST   I   CLIVE   SYS  02-09-19 18:40 DP:   Failed
   JOB ADAPT03 ST   I   CLIVE   SYS  02-09-19 18:55 DP:   Staged
   JOB ADAPT04 CONS P   CLIVE   SYS  02-09-19 18:44
   JOB ADAPT05 ST   I   CRAIG   SYS  02-09-23 14:14
   JOB JCL001 ST   I   CLIVE   SYS  02-09-19 18:34 DP:   Failed
   CLST REXX01 ST   I   CRAIG   SYS  02-09-04 08:55 SP:   S000002449
   CLST REXX02 ST   I   CRAIG   SYS  02-09-25 06:37
   CLST REXX03 ST   I   KARYNS  SYS  02-09-25 08:52
   CLST REXX04 ST   I   CRAIG   SYS  02-09-25 06:37
   CLST REXX05 ST   I   CRAIG   SYS  02-09-25 06:37 SP: I   Held
   CLST REXX06 ST   I   CRAIG   SYS  02-09-25 06:37
   CLST REXX07 ST   I   CRAIG   SYS  02-09-25 15:47 DP:   Staged
   CLST REXX08 ST   I   CRAIG   SYS  02-09-25 15:55 DP:   Implement
   CLST REXX09 ST   I   CRAIG   SYS  02-09-25 07:27
    
```

DT Screen

Entering the DT operation against a Task that has participated in Deployment displays the screen shown in [Figure 91](#).

Figure 91 DT Screen

```

ISPW 18.02          DEPLOY TASK STATUS          Row 1 of 4
Command ==>          Scroll ==> CSR
                                More ==>

  DplyType DplyCat  DplyEnv  Sub  System  PartName Status  Appl
-----
JCL BATCH  ACTST  CT   SYS-FB  ADAPT03 Confirmed  SYS
JCL BATCH  ACTST  CT   SYS-FA  ADAPT03 Confirmed  SYS
JCL BATCH  ACTST  II   SYS-FB  ADAPT03 Confirmed  SYS
JCL BATCH  ACTST  II   SYS-FA  ADAPT03 Confirmed  SYS
-----
Bottom of List -----
    
```

[Figure 91](#) shows all the Physical Packages with their statuses. This screen can be scrolled to the right.



If the Task is not currently participating in Deployment, Package details are displayed for the most recent Deployment at the current level.

Columns

The columns on the Task Status screen are listed [Table 57](#).

Table 57 Task Status Columns

Column	Description
DplyType	Deploy Type
DplyCat	Deploy Category
DplyEnv	Deploy Environment
Sub	Deploy Sub-environment

Table 57 Task Status Columns (*Continued*)

Column	Description
System	Target system that this Physical Package is deploying to.
PartName	Component Part Name
Status	Physical Package Status. See the sections above for a list of all possible values.
Appl	ISPW Application
Storage Name	This is the Dataset Name/HFS Path that the Part is to be deployed to.
Date/Time Impl	This is the Date and Time constraint for the Implementation process.
Date/Time Actv	This is the Date and Time constraint for the Implementation process.
ReqlD	Deploy RequestID
PkgID	PackageID
ItmID	Unique ItemID of this component Part within the Package
PrtID	Unique ID for the Component Part within the Deploy Request.

Column Display

As with most ISPW list screens, the VP (View Preferences) command can be used to tailor the display to the User’s preference.

Operation History – Deploy Operations

Historical Deploy Request information can be displayed by entering the DP and DT operations against the Operation History Display.

Operation History

The Operation History is displayed by entering H against a Task. All logged operations will be displayed ([Figure 92](#)).

Figure 92 Task Event History Screen

```

ISPW 18.02          TASK EVENT HISTORY          Row    1 of   16
Command ==>                               Scroll ==> CSR

  Appl: ABN  Type: CLST Name: REXX02  Clas:

  Op A Lev  Targ User      Date      Time  Set
-----
  I  ST  ST  CRAIG    2003-08-25 20:07 S000003427
  I  ST  ST  CRAIG    2003-08-25 20:06 S000003426
  I  ST  ST  CRAIG    2003-08-25 20:02 S000003425
  I  ST  ST  CRAIG    2003-08-25 18:52 S000003424
  I  ST  ST  CRAIG    2003-08-25 18:50 S000003423
  I  ST  ST  CRAIG    2003-08-25 18:49 S000003422
  I  ST  ST  CRAIG    2003-08-25 18:37 S000003421
  I  ST  ST  CRAIG    2002-09-26 07:49 S000002624
  I  ST  ST  CRAIG    2002-09-25 06:37 S000002599
  I  ST  ST  CRAIG    2002-09-25 06:12 S000002598
  I  ST  ST  CRAIG    2002-09-23 14:11 S000002579
  I  ST  ST  CRAIG    2002-09-19 14:13 S000002553
  I  ST  ST  CRAIG    2002-09-19 14:12 S000002552
  P  CONS ST  CRAIG    2002-09-19 14:12 S000002552
  P  UT  CONS CRAIG    2002-09-19 14:11
  C  UT  UT  CRAIG    2002-09-10 02:03
----- Bottom of List -----
    
```

Each of the I operations above represented a deploy, so entering either DP or DT will display the relevant Deploy information for that historical Deployment.

Deploy Set Processing

This section explains how ISPW Deploy interacts with Set Processing.

Dates and Times

When ISPW Sets are created, three date/time values are specified as listed in [Table 58](#).

Table 58 ISPW Sets Date/Time

Date/Time	Description
Set Start	The Time that the Set will be dispatched for execution.
Implementation	The earliest time that the Deploy Implementation process can begin. Staging can occur before this time.
Activation	For defined activation processes, this is the earliest time that they can be dispatched for execution.

Some or all of these may or may not be on the “Set Create” screen at your site. This depends upon the processes defined for the setting of these at your particular site.

New Set Deploy Status

The Set Status incorporates the Deployment Status and can be the following values:

Table 59 Set Status

Status	Description
Confirmed	All Deploy Requests have been Confirmed
Staged	Staging is complete for all Deploy Requests
Implemented	Implementing is complete for all Deploy Requests – Activation is still required on at least one.
Completed	All Deploy functions are complete.
Failed	One or more Deploy Requests has Failed.

These new statuses are prefixed by DP: in the Task Status area ([Figure 87](#) on page 94) or suffixed by the word “Deploying” on the Set Modify screen ([Figure 93](#)).

Figure 93 Set Modify Screen

```

ISPW          MODIFY/APPROVE SET S000002611          Row 1 of 1
COMMAND ==>

Description ==> TEST DEPLOY
Owner       ==> CRAIG
Change Type ==> STANDARD CHANGE
Operation   ==> I
Level      ==> ST
Build      ==> NO
Release    ==>
Create     ==> 2002-09-25 15:55:04
Start      ==> 2002-09-25 15:55:00
Implement  ==> 2002-09-25 15:55:00
Activate   ==> 2002-09-25 15:55:00
Queue      ==>
Job/Proc   ==> / WZDW3
Class      ==> A
Lock Status ==> LOCKED
Aprv Status ==> NOT REQUIRED
Exec Status ==> IMPLEMENTED - DEPLOYING

Set Control ==> (BROWSE)
----- APPROVALS -----

Enter A to approve, D to deny or P to reset to pending and press Enter

CMD  APPROVER SEQ  REQD  CODE  USERID  DATE/TIME
OPS   1      N    PENDING
***** Bottom of data *****

```

Deploy Logging

Introduction

Intended Audience

It is assumed that the reader has a basic knowledge of ISPW Deploy and how to make ISPW customizations.

ISPW Services

Some local configuration may be required. If externally sourced resources are preferred to perform this work, ISPW Services can be arranged.

Background

ISPW's z/OS Deploy was originally developed to support and manage the Fully-Populated Deploy model, with the requirement of deploying cross platform modules without the need for shared DASD, the standard in Production environments. In addition, ISPW z/OS Deploy also provides support of the Concatenated model.

In Test environments, although the Fully-Populated Deploy model is used, many customers use the Concatenated model instead—where the life cycle datasets are concatenated into the runtime environments. Both Base and z/OS Deploy ISPW products support the Concatenated model as part of its standard life cycle support and as software parts (LOAD module, DBRM, etc.) are promoted up to the next testing level, they are deleted from the one behind.

Overview

ISPW's z/OS Deploy capabilities are increasingly being used for test environments within the same LPAR where the Concatenated model is required. Deploy is required because:

- ISPW Deploy's superior functionality around coordinated and controlled implementations is useful even for Test environments.
- Many sites now need the "Selective Deploy" capability where at a single level the software parts might need to be deployed to any number of target environments.

Deploy's native support for the Fully-Populated model has led to over-complicated hard-coded logic required in the deploy scripts and/or over-complicated combinations of deploy environments and sub-environments to perform clean ups of libraries.

Features

A Deploy Logging feature has been developed to better support the Concatenated model in situations using Deploy.

It includes:

- A new DB2 Repository Tablespace, Table and associated Indexes

- Three new WZZTSI calls to interact with the data (List, Add, Cleanup).

Configuration

M.ER

Define a new M.ER variable called DPRTINST and set the value to Y. Refresh the server.

New TSI Calls

There are three new TSI calls available to use in custom Deploy code:

- DPLYPRTI, LIST – List Deploy Part Instance Log Entries
- DPLYPRTI, ADD – Add Deploy Part Instance Log Entry
- DPLYPRTI, CLEANUP – Clean Up Deploy Part Instance Log Entry.

DPLYPRTI LIST

This call can be made at some point during the Promotion process (usually in the Set Post Exit) to list all the Target runtime members and libraries that have been deployed to by the Task being promoted. It details all the Deploy Part Instances that have been deployed for that Task from an Application Stream Level in the ISPW lifecycle.

[Table 60](#) describes the input and [Table 61](#) describes the output fields.

Table 60 DPLYPRTI LIST Input Fields

Input Field	Description
TASKIDX,LENGTH=12	Task ID in hex format
APPLID,LENGTH=4	The Application Level in the ISPW Lifecycle where Deploys are initiated
STRMNAME,LENGTH=8	
LCLVL,LENGTH=4	

Table 61 DPLYPRTI LIST Output Fields

Output Field	Description
DPRTIID,FORMAT=INTEGER,TYPE=KEY	Deploy Part Instance ID
DPARTID,FORMAT=INTEGER	Deploy Part ID
DPRTINAM,LENGTH=128	Deploy Part Instance name
DPTYPE,LENGTH=8	Deploy Type
PARTTYPE,LENGTH=4	Part Type
PARTCLAS,LENGTH=4	Part Class
PARTNAME,LENGTH=64	Part Name
TASKID,LENGTH=6,FORMAT=BINARY	Task ID
CMPNID,LENGTH=6,FORMAT=BINARY	Component ID
CMPVID,FORMAT=SMALLINT	Component VID
DREQID,FORMAT=INTEGER	Deploy Request ID
PKGID,FORMAT=SMALLINT	Deploy Package ID
ITEMID,FORMAT=SMALLINT	Deploy Item ID
SYSTNAME,LENGTH=8	Deploy System Name
DPENV,LENGTH=8	Deploy Environment

Table 61 DPLYPRTI LIST Output Fields (*Continued*)

Output Field	Description
SUBENV,LENGTH=4	Deploy Sub-Environment
STORTYPE,LENGTH=4	Storage Type
STORNAME,LENGTH=100	Storage Name
STORUSGE,LENGTH=1	Storage Usage
CTSRVRNM,LENGTH=8	CT Server Name
ACTIDTTM,LENGTH=26	Deploy Log Entry Timestamp

DPLYPRTI ADD

This new ISPW call will add a Deploy Part Instance Log Entry. It is not required when the Deploy Implementation Type has been set to C (the Copy to the Target is performed internally by ISPW), because the Deploy Logging happens automatically. It is only required when the ITEM, GET call is being used to copy to the Target. In this situation, Deploy Logging must be requested manually using this new DPLYPRTI, ADD call.

[Table 62](#) describes the input fields.

Table 62 DPLYPRTI ADD Input Fields

Input Field	Description
DPRTINAM,LENGTH=128	Deploy Part Instance Name (optional), defaults to Deploy Part Name if not provided.
DIDREQID,FORMAT=INTEGER,VNAME=DREQID	Deploy Request ID
DIPKGID,FORMAT=SMALLINT,VNAME=PKGID	Deploy Package ID
DIITEMID,FORMAT=SMALLINT,VNAME=ITEMID	Deploy Item ID
DISYSTNM,LENGTH=8,VNAME=SYSTNAME	Deploy System Name
DISTORTP,LENGTH=4,VNAME=STORTYPE	Deploy Part Instance Storage Type, Name and Usage. If these fields aren't supplied on input, they are looked up from the Deploy Logical Storage Group configuration definition.
DISTORNM,LENGTH=100,VNAME=STORNAME	
DISTORUS,LENGTH=1,VNAME=STORUSGE	

DPLYPRTI CLEANUP

This new ISPW call can be used when deleting members from the Target runtime libraries. It tracks this event by updating the current Deploy Part Instance Log Entry to become inactive.

[Table 63](#) describes the input fields.

Table 63 DPLYPRTI CLEANUP Input Fields

Input Field	Description
DPRTIID,FORMAT=INTEGER	Deploy Part Instance ID

Sample Code

The code in [Figure 94](#) on page 102 shows an example Clean Up stub in the Set Post Exit. In the standard ISPW delivered sample it could reside in WZUSETX.

Figure 94 Sample Code – Cleanup_Deploy

```

/* ----- */
/* This function is called on a promote to cleanup the datasets
/* deployed to at the previous level
/* ----- */

call Open_Set
"ISPEXEC TBSKIP ASTABLE"
tbrc = rc

Do While tbrc = 0 DZTABLE =
"DZTABLE"
taskidx=c2x(taskid) STRMNAME =
sxstrm LCLVL = oplvl
retc = WZZTSI("DPLYPRTI","LIST")
say "DPLYPRTI LIST rc = "retc Address ISPEXEC
If retc = 0 then do "TBTOP"
DZTABLE
Do Until skiprc > 0 "TBSKIP"
DZTABLE
skiprc = rc
If rc = 0 then do
say DPRTIID DPRTINAM STORTYPE STORNAME STORUSGE CTSRVNM
delpds = stornam delmem =
dprtina Call Delete_Member
retcu = WZZTSI("DPLYPRTI","CLEANUP")
say " DPLYPRTI CLEANUP rc = "retcu End
End
"TBEND" DZTABLE
End
"ISPEXEC TBSKIP ASTABLE"
tbrc = rc End

```

Processing Details

This section describes the events that occur during normal processing for Deploy Logging.

DB2 Table

The ISPW DB2 Table (WZT_DP_PART_INSTANCE) stores the details of all the deployed Part Instances for a Task in the runtime libraries on an ISPW CT Server. [Table 64](#) describes the columns.

Table 64 New DB2 Table

Column	Description
PART_INST_ID	Deploy Part Instance ID
PART_ID	Deploy Part ID
PART_INST_NAME	Deploy Part Instance Name
REQ_ID	Deploy Request ID
PKG_ID	Deploy Package ID
ITEM_ID	Deploy Item ID
SYST_NAME	Deploy System Name
APPL_ID	Foreign key for the M.AD(L) entry representing the ISPW Lifecycle level where the Deploy was initiated
STREAM_NAME	
LC_LVL	
DP_ENV	Deploy Environment
SUB_ENV	Deploy Sub-Environment

Table 64 New DB2 Table (*Continued*)

Column	Description
STORAGE_TYPE	The details of the runtime library where the Part Instance has been deployed
STORAGE_NAME	
STORAGE_USAGE	
CT_SRVR_NAME	ISPW CT Server Name
TASK_ID	Task ID
CMPNT_ID	Component ID
CMPNT_VID	Component VID
CURR_DPLY_INST	Values can be: C – current Deploy Part Instance for the runtime library on that CT server H – historical entry due to either a subsequent Deploy creating a new Deploy Part Instance or a Clean Up was requested
HIST_STATUS	Values can be: Blank – current S – Superseded (replaced by a subsequent Deploy) D – replaced by an Action D request N – null entry for new Action D requests C – Clean Up was performed
INST_TYPE	Internal use
ACTIVE_TSTAMP	Time when the Deploy Part Instance was created
INACTIVE_TSTAMP	Time when the Deploy Part Instance became historical by being replaced, deleted or cleaned up

Add Processing

When a Task is deployed, a new Deploy Log Entry will be added for each Instance of a Deploy Part in the target runtime libraries. This results in a new row in the DB2 table, which relates the Deploy Part Instance to the ISPW Deploy Part and the Deploy Request.

[Table 65](#) describes the status values for the new row.

Table 65 Add Processing

Column	Value
PART_INST_ID	Generate a new IDENTITY integer
CURR_DPLY_INST	C
HIST_STATUS	<space>
ACTIVE_TSTAMP	Current date and time
INACTIVE_TSTAMP	null

Subsequent Deploys

When a Task is deployed again where there is still a current Deploy Part Instance with the same Name in the same runtime library (as specified in Storage Type/Name on that CT Server), the existing Deploy Log Entry will be updated to become **Historical**. The status is updated to **Superseded** and the Inactive Timestamp is set to the current time.

[Table 66](#) describes the status values for the updated existing row.

Table 66 Subsequent Deploys

Column	Value
CURR_DPLY_INST	H
HIST_STATUS	S
INACTIVE_TSTAMP	Current date and time

A new Deploy Log Entry is then created to reflect the new Deploy Part Instance that was deployed, as detailed in [Add Processing](#) on page 103.

Action D Processing

If the Task being deployed is an Action D task and there is an existing Deploy Log Entry, it will be updated to become Historical. The status is updated to Deleted and the Inactive Timestamp is set to the current time.

[Table 67](#) describes the status values for the updated existing row.

Table 67 Action D Processing

Column	Value
CURR_DPLY_INST	H
HIST_STATUS	D
INACTIVE_TSTAMP	Current date and time

A new Deploy Log Entry is then created to reflect the new Deploy Part Instance that was deleted, however it is created as an Historical entry with a Null status so it can be linked back to the Deploy Request.

[Table 68](#) describes the status values for the new row.

Table 68 Status Values

Column	Value
PART_INST_ID	Generate a new IDENTITY integer
CURR_DPLY_INST	H
HIST_STATUS	N
ACTIVE_TSTAMP	Current date and time
INACTIVE_TSTAMP	null

Clean Up Processing

Deploy Log Entries can be made inactive by requesting a Clean Up. They are then eligible for deletion at a later time.

Table 69 Clean-Up Processing

Column	Value
CURR_DPLY_INST	H
HIST_STATUS	C
INACTIVE_TSTAMP	Current date and time